WebOTX トラブルシューティング(概要)

WebOTX 運用編

バージョン: 7.1

版数: 第 8 版

リリース: 2014 年 3 月

目次

1. はじめに

本書は WebOTX 実行環境を運用するための運用操作法について概要や具体的な設定項目や設定方法について記載しています。

対象読者

このマニュアルは WebOTX Application Server Web Edition、Standard-J Edition、Standard Edition、Enterprise Edition を使って運用環境を構築するシステムエンジニア、日々の運用を行うオペレータを対象としています。

表記について

パス名表記

本書ではパス名の表記については特に OS を限定しない限りセパレータはスラッシュ'/'で統一しています。Windows 環境においては'¥'に置き換えてください。

環境変数表記

インストールディレクトリやドメインルートディレクトリなど環境によって値の異なるものについては環境変数を用いて表します。

${*env*} または $(*env*)で表しています。

例)
${AS_INSTALL} : インストールディレクトリ
${INSTANCE_ROOT}: ドメインルートディレクトリ

製品名表記

以後の説明で「WebOTX」としているものは、「WebOTX Application Server」のことをあらわします。

コマンド操作について

本書中では運用操作に用いるコマンドの詳細についての説明は省略しています。

コマンドの詳細は「運用管理コマンド」、「運用管理コマンドリファレンス」を参照してください。

2.障害解析

障害発生時採取する情報について説明します。

2.1. 全般

障害が発生した場合、障害内容にかかわらず採取すべき情報について説明します。

エージェントログファイル
　${ INSTANCE_ROOT}/logs ディレクトリのファイル全て

イベントログ情報(UNIX はシスログ)
　(Windows) イベントログ(アプリケーション、システム)
　(HP-UX) /var/adm/syslog/syslog.log
　(Solaris) /var/adm/messages
　(Linux) /var/log/messages

プロセス異常終了情報
　(Windows)
　ワトソンログ %SystemDrive%¥Documents and Settings¥All Users¥Documents¥DrWatson
　クラッシュダンプ %SystemDrive%¥Documents and Settings¥All Users¥Documents¥DrWatson¥user.dmp
　(UNIX) core ファイル

UNIX 版における core ファイルの出力先は以下のようになります。

サービス名	プロセス名	出力場所	備考
WebOTX エージェント	java (agent)	${INSTANCE_ROOT}/config	
Web サーバ(Apache)	httpd	${INSTANCE_ROOT}/	
ObjectBroker サービス	oad	${INSTANCE_ROOT}/config	
	namesv	${INSTANCE_ROOT}/config	
	cnamesv	${INSTANCE_ROOT}/config	Enterprise Edition のみ
	java (oadj)	${INSTANCE_ROOT}/config	
	ospprxy	${INSTANCE_ROOT}/config	Enterprise Edition のみ
JMS サービス	java (jms)	${INSTANCE_ROOT}/config	
Transaction サービス	rcssv	${INSTANCE_ROOT}/config	Standard Edition/ Enterprise Edition のみ
TP システム	iioplsn	${AS_INSTALL}/Trnsv/logs	Standard Edition/ Enterprise Edition のみ
	tpmonitor	${AS_INSTALL}/Trnsv	Standard Edition/ Enterprise Edition のみ
	THTPPJAVA2 THTPPCTL<数字>	${AS_INSTALL}/Trnsv	Standard Edition/ Enterprise Edition のみ

	THTPPOTS<数字> THTPPDB<数字>		<数字>には 9,8,7 のいずれかが入ります
TP モニタ制御サービス	tpadmd	${AS_INSTALL}/Trnsv	Standard Edition/ Enterprise Edition のみ

WebOTX Standard Edition/Enterprise Edition での障害を未然に防ぐために知っておきたい知識や、障害発生時の原因究明に役立つ機能について、2.1.1 以降に記述しています。

2.1.1. アプリケーションプロセス異常終了に対する機能

WebOTX Standard Edition/Enterprise Edition では TP モニタ機能により、アプリケーションプロセスの状態を常に監視しています。アプリケーションプロセスが異常終了した場合、そのアプリケーションプロセスは自律的に復旧されます。またその際に、詳細な障害情報を記録しますので、後からの障害解析を容易に行うことができます。

2.1.1.1 異常終了の検出と自律復旧

アプリケーションプロセスの異常終了を検知し、そのプロセスが使用していたリソース(コネクション、共有メモリ、データベース資源)を WebOTX が解放します。さらに指定に従ってプロセスの再起動を行うことができます。これにより、障害発生後もそれ以前と同じ処理能力を維持することができます。

アプリケーションプロセスの異常終了が検出されると、イベントログ・シスログに以下のようなメッセージが出力されます。**異常が発生した場合はまずはイベントログ・シスログを確認してください。**

> OTXM:<システム名>::<プロセス ID>: W:13:TPS15-01107 Process abnormal end. PID=[<プロセス ID>], class[<プロセスグループ名>], ped[<アプリケーショングループ名>.ped]

プロセスグループが複数プロセスで構成される場合、その中の一つのアプリケーションプロセスが異常終了しても、他のアプリケーションプロセスは通常通り処理を行うことができます。このため、プロセスグループとしては、異常終了したアプリケーションプロセスが再起動されるまでの間もサービスの継続性を確保できます。

アプリケーションプロセスの再起動回数を設定することができます。プロセスの異常終了があるとこの指定に応じて再起動されます。ただし一度障害が発生した後に自動的に再起動させても同じ障害の繰り返しとなる可能性もありますので注意が必要です。また自動的に再起動させてしまうと障害が起きたことに気づかない恐れがあるので、テスト環境で使用する場合は再起動させない設定とし、本番運用では再起動する設定とすることを推奨します(統合運用管理ツールの[TP システム]-[上限設定]-[プロセス障害時の再起動回数])。

C++で作成されたサーバアプリケーションの場合は、異常終了発生後にオペレーションを閉塞状態にしてクライアントから実行させないようにすることができます。アプリケーション作成者は、閉塞している間にプログラムのチェックを実施することになります。オペレーション単位で閉塞状態にすることができるので、正常動作している他のオペレーションは引き続き動作させることは可能です。閉塞されたオペレーションを呼び出すと以下のメッセージがイベントログ・シスログに出力され、問題のオペレーションは実行されません。

> OTXM:<システム名>:<プロセスグループ名>:<プロセス ID>: E:1:TPS10-03201 OPERATION BLOCKED. TX-GROUP プロセスグループ名 プロセス ID

オペレーションの閉塞に関する設定は統合運用管理ツールのプロセスグループの[スレッド制御]-[オペレーション異常終了時にオペレーションを閉塞させる]で行います。

なお、V5 までは閉塞するのがデフォルトの設定となっていましたが、V6 以降では閉塞させないのがデフォルトの設定と変わりました。

2.1.1.2 リクエスト保障

オペレーション実行中に何らかの障害が発生してプロセス終了となった場合、TP モニタがそのプロセスに代わりエラー応答をクライアントに返します。イベントログ・シスログにも情報を残します。キューで待っていたリクエストは、別プロセスが引き継いで処理をすることができます。他に起動中のアプリケーション

プロセスがなく、かつプロセス再起動回数を使い果たした場合は別プロセスに引き継げませんが、この場合はエラー応答をクライアントアプリケーションに返すため、サーバアプリケーションの異常終了に引きずられてクライアントアプリケーションが無応答になることはありません。

クライアントアプリケーションに返却されるエラーコードの詳細は「メッセージ編」を参照してください。

2.1.1.3 障害情報記録(Java)

オペレーション実行中に Java ランタイム例外が発生した場合、正確には WebOTX の基盤で Java ランタイム例外を捕捉した場合はアプリケーションプロセスを異常終了させますが、この際に捕捉した例外のスタックトレースがシステムトレースに記録されます。この情報から、Java ソースのどの行が原因で異常終了したかを特定することができます。

デフォルトでは以下の場所にアプリケーションログを出力しています。
${ INSTANCE_ROOT}/logs/tpsystem/(アプリケーショングループ名)/(プロセスグループ名)
なお異常終了してしまった場合は save ディレクトリにログが移動しますので save ディレクトリも確認ください
ファイル名:
(プロセスグループ名).(時間をベースとした識別子).(プロセス ID).log

例:

2005/07/15 10:23:53.125|001: Error: Occur exception(catch java.lang.RuntimeExcep

tion) at Invoking. message=null

java.lang.RuntimeException

at complex.basicApObj.exceptionEnd(basicApObj.java:21)

at complex.basicApPOA._invoke(basicApPOA.java:55)

・・・

2.1.1.4 障害情報記録(C++)

オペレーション実行中にネイティブ例外が発生した場合、正確には例外ハンドリングを行う設定(C++デフォルト)にしたアプリケーションの WebOTX の基盤でネイティブ例外を捕捉した場合は該当スレッドを閉塞させます。スレッドの残りがない場合はプロセスを終了させます。ネイティブ例外を捕捉した際はイベントログ・シスログにメッセージを出し、ネイティブ例外エラーコードも共に出力します。

さらにシステムトレースに記録されたライブラリロードアドレスと開発時の map ファイルを使用することにより、障害箇所を特定できる可能性があります。

例外発生時の情報はアプリケーションログに出力されます。デフォルトでは以下の場所にアプリケーションログを出力しています。
${ INSTANCE_ROOT}/logs/tpsystem/(アプリケーショングループ名)/(プロセスグループ名)
なお異常終了してしまった場合は save ディレクトリにログが移動しますので save ディレクトリも確認ください
ファイル名:
(プロセスグループ名).(時間をベースとした識別子).(プロセス ID).log

```
2005/12/08 16:47:34.283|001: Error: AbortExit(exitkey=82, endkey=00).
2005/12/08 16:47:34.283|001: Error: Exception occurred. repositoryID=IDL:demo3:1.0 operation=AbortCall3
2005/12/08 16:47:34.283|001: Error: AbortExit(exitkey=82, endkey=00)(finished).
Thu Dec 08 16:47:34 2005 TPS10-13201 TPabortによるプロセス終了 CODE:0 TX-GROUP demo3PG 11156
Thu Dec 08 16:47:35 2005 TPS10-13201 TPabortによるプロセス終了 CODE:0 TX-GROUP demo3PG 11156
```

アプリケーションログにはアプリケーションエラーが発生した旨のエラーメッセージを出力します。これに

よりどのオペレーション呼び出しでエラーが発生したか特定することは可能ですが、Java のようにスタックを出力することはできません。スタックイメージを確認するには例外ハンドルの設定を無効にし、ワトソンログ出力を有効にする必要があります。

例外ハンドルの設定は統合運用管理ツールの[TP システム]-[例外ハンドル]-[例外ハンドルを行う]です。

2.1.1.5 異常終了時のオペレーション状態記録

アプリケーションプロセスが異常終了時に、実行中だった未完のオペレーションの状態がシステムトレースに記録されます。この情報から、どのオペレーション実行中に異常終了したかをオペレーション単位で絞り込むことができます。本情報がなくとも前記 2.1.1.3、2.1.1.4 の情報で十分な場合もありますが、2.1.1.3, 2.1.1.4 で説明した機能はアプリケーションプロセス中で実行される機能であるため異常の程度によっては機能しないことがあります(例えば kill コマンドやタスクマネージャで強制終了させた場合)。これに対し、本機能はアプリケーションプロセスが実行の都度共有メモリに記録している情報を元に出力しているので、あらゆる終了に対応できます。

[出力例]

以下のログは 7/15 10:19:09.703 に開始したオペレーション(オペレーション名:LoopBack, インタフェース名:IDL:sample/LoopBackSample:1.0, モジュール名:loopback_jsv.jar)が実行している最中にアプリケーションプロセスが異常終了したことを示します。

```
Error:   The following TX execution is unfinished.
----------------------------------------
PID     ThID   TxID          StartTime
----------------------------------------
03761 00001 QKAAAC        7/15 10:19:09.703
                  QKAAAC:     LoopBack (IDL:sample/LoopBackSample:1.0;loopback_jsv.jar)
```

2.1.2. 無応答障害に対する機能

Standard Edition/Enterprise Edition では、アプリケーションプロセスが無応答や極度の遅延に陥った場合も、それを検出し、情報を採取し、リカバリを行う機能を実装しています。

2.1.2.1 プロセス起動停止の実行時間超過の検出および全スレッドストールの検出

プロセス起動処理(Java VMの起動,ORB初期化など)、プロセス終了処理(ORB解放,Java VM停止など)の無応答障害をAPループ検出機能により検出し、該当アプリケーションプロセスは強制終了します。起動処理の障害で且つ再起動設定があれば再起動されます。また、次の 2.1.2.2 や 2.1.2.3 で説明する監視機能はアプリケーションプロセス中の制御スレッドで実装されていますが、この制御スレッドさえも正常に機能しないような異常状態になった場合も、APループ検出機能により障害を検出し、アプリケーションプロセスを強制終了します。タイマの設定方法は「運用編(チューニング) 2.5.2 起動・停止タイムアウト(プロセス終了)」を参照してください。

2.1.2.2 スレッド起動停止の実行時間超過の検出

スレッド起動処理(サーバオブジェクトや常駐オブジェクト作成)、スレッド終了処理(サーバオブジェクトや常駐オブジェクト削除)の際の無応答障害を検出してプロセスは強制終了します。起動処理の障害で且つ再起動設定があれば再起動されます。スレッド初期化タイムアウト時間はプロセスグループ単位に設定できます。タイマの設定方法は「運用編(チューニング) 2.5.2 起動・停止タイムアウト(プロセス終了)」を参照してください。

2.1.2.3 オペレーション実行時間超過の検出と自律復旧

サーバアプリケーションでオペレーションを実行した際の無応答障害を検出し、自律復旧させることができます。

実行時間上限設定を超えてもサーバアプリケーションから応答がない場合、WebOTX はアプリケーションプロセスが無応答障害に陥っていると判断し、アプリケーションプロセスを強制終了し、再起動設定が

あれば再起動させます。これによりクライアントに応答が返らない事態やサーバの資源を無制限に消費する事態を回避できます。

実行時間上限値はオペレーション毎に設定することができます。設定方法は「運用編(チューニング) 2.5.4 実行時間タイムアウト」を参照してください。

実行時間上限設定を適切に行うために、運用アシスタントの実行時間上限の適正値算出機能を利用することができます。詳細は「運用編(TP モニタの運用操作) 2.40.4 実行時間上限の適正値算出」を参照してください。

また、データベースのデッドロック等、再試行可能な障害の発生も考えられます。その場合に、そのオペレーションを自動的にやり直したり、データベースがオーバーフローしたりした場合には、更新系のサービス(オペレーション)だけを停止する等、障害発生時にきめ細かいリカバリ処理を 行うこともできます。

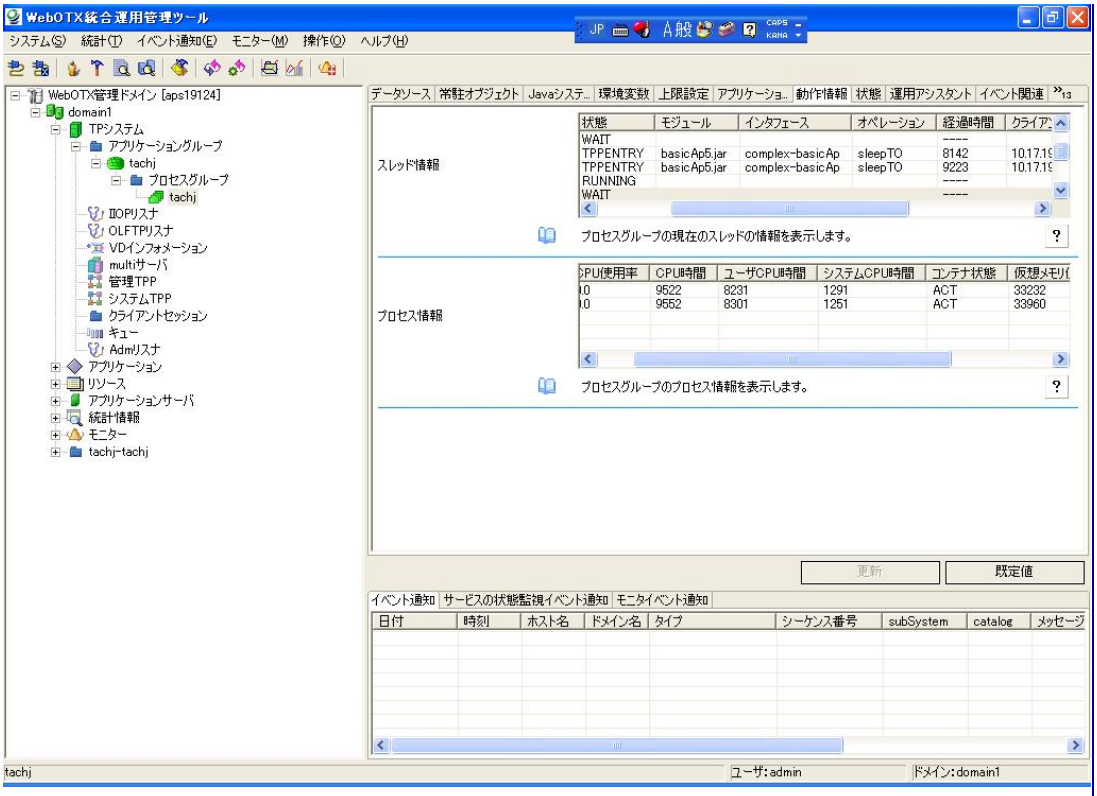
2.1.2.4 リクエスト保証(AP 応答監視タイマ)

AP 応答監視タイマ機能によりサーバアプリケーションの無応答障害を検出し、別プロセスが異常状態のアプリケーションプロセスに成り代わってクライアントアプリケーションにエラー応答(NO_RESPONSE(3920))を返すことができます。これによりサーバアプリケーションの無応答障害が発生しても、クライアントアプリケーションは無応答状態とならずに応答を受け取ることができます。

AP応答監視タイマ機能はプロセスグループ毎に設定することができます。設定方法は「運用編(チューニング) 2.5.3 呼び出しタイムアウト」を参照してください。

2.1.2.5 スレッド状態表示

WebOTX ではサーバアプリケーションの全スレッドでどのような処理が行われているかの情報を取得することができます。各スレッドの状態(IDLE 状態/実行状態など)、実行中のオペレーション名、実行中オペレーションの開始時刻、接続しているクライアントアプリケーションの IP アドレスが可視化されており、各スレッドの詳細情報を確認できます。無応答障害が発生している最中にこの情報を確認することで、障害原因を絞り込むことができます。また、各スレッドの CPU 時間を確認することで、CPU ループを引き起こしているオペレーションを特定することもできます。
* Linux 版、Solaris 版では各スレッドの CPU 時間を採取できません
また、各プロセスの CPU 使用率、CPU 使用時間(ユーザモード/システムモード)、メモリ使用量(仮想/物理)コンテナ状態(起動処理の進み具合)についても採取することができます。
* Linux 版では CPU 情報を採取できません



2.1.2.6 障害情報記録(スタック採取)

無応答障害に陥っているアプリケーションプロセス(Java)のスタックトレースを採取することができます。この情報より、サーバアプリケーション中のどの箇所で無応答となっているかを特定することができます。上記の実行時間上限監視により無応答障害を検出すると、このスタックトレースを採取し AP トレースに記録します。

例:

2005/07/15 10:23:53.125|001: Error: Occur exception(catch java.lang.RuntimeExcep

tion) at Invoking. message=null

java.lang.RuntimeException

at complex.basicApObj.exceptionEnd(basicApObj.java:21)

at complex.basicApPOA._invoke(basicApPOA.java:55)

・・・

無応答障害発生中に、運用管理者がコマンドwostack(「運用コマンドリファレンスマニュアル」→「CORBAアプリケーション」→「wostack」)を使用することにより、任意のタイミングでスタックトレースをAPトレースに記録することも可能です。このコマンドによりアプリケーションを停止させるなどの影響はありません。

また、実行時間上限監視により無応答障害を検出すると、終了直前に実行中だった未完のオペレーションの状態がシステムトレース(運用編(ロギング) 2.5.1 サーバアプリケーショントレース採取)に記録されます。この情報から異常終了の原因をオペレーション単位で絞り込むことができます。

例:

Error: The following TX execution is unfinished.

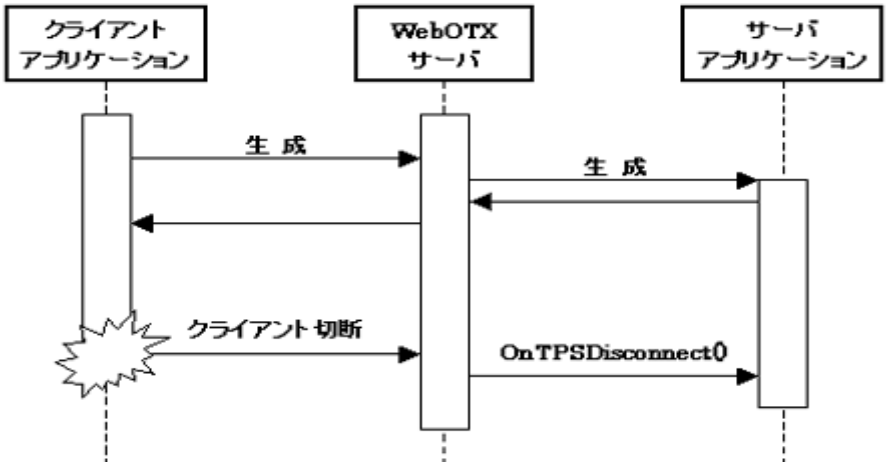
```
PID     ThID    TxID            StartTime
---------------------------------------
03761 00001 QKAAAC          7/15 10:19:09.703
                    QKAAAC:     LoopBack (IDL:sample/LoopBackSample:1.0;loopback_jsv.jar)
```

2.1.3. ネットワーク/クライアント障害に対する機能

2.1.3.1 通信の監視(無通信監視)

接続しているクライアントアプリケーションより、一定時間が経過してもオペレーション呼び出しがない場合、強制的にクライアントからの接続を切断させることができます。これにより、長時間使用されていない不必要と思われる接続が切断され、ネットワークリソースの消費を抑えることができます。設定方法は「運用編(チューニング) 2.5.5 クライアント無通信監視間隔」を参照してください。

また、WebOTX で提供する、コールバック用の API を利用すると、CORBA サーバアプリケーションがクライアント障害の発生の通知をシステムから受け取って、独自の後処理を行うことができます。



2.1.3.2 アライブチェックによる監視

TCPレベルのkeepalive機能を設定することにより、アライブチェックを行うことができます。これによりネットワーク障害で使用不可能となってしまった接続を検出し切断することができます。 keepalive機能自体はOSが持つ監視機能となります。WebOTXではOSのkeepalive機能使用有無のみを設定します。keepalive機能のアライブチェック間隔は各OSに依存します。WebOTXに対する設定は「クライアント接続数オーバへの対応」を参照してください。

2.1.4. スローダウン障害に対する機能

2.1.4.1 スローダウン障害検出

オペレーションの呼び出し時間の統計より、通常よりオペレーション実行時間が長くなっている場合、スローダウンとして検出し通知を行ないます。一般的な監視機構とは違い、オペレーションごとの閾値設定なしでスローダウン障害を検出できます。

スローダウン発生時は、イベントログ・シスログに以下のメッセージが出力されます。
“OTX20110100 オペレーションzzzのスローダウンを検出しました。current:平均実行時間=xxx 秒。normal:平均実行時間=www 秒。プロセスグループ=vvv。 ObjectName=yyy”
“OTX20110100 The Operation zzz get late. Average of current time is xxx s. Average of normal time is www s. The Process Group is vvv. The ObjectName is yyy“.

オペレーションのスローダウンを検出してから「スローダウン継続監視時間」を超えてなお、スローダウン状態が継続していると以下のメッセージを統合運用管理ツールに通知し、イベントログ/syslog,webotx_agent.log(webotx_tpmmgr.log)に出力します。

“OTX20110200 オペレーションzzzの長期にわたるスローダウン状態を検出しました。current:平均実行時間=xxx 秒。normal:平均実行時間=www 秒。スローダウン継続時間=uuu 分。プロセスグループ名=vvv, ObjectName=yyy”
“OTX20110200 The Operation zzz is slow for a long time. Average of current time is xxx s. Average of normal time is www s. Duration of slow is uuu m. The Process Group is vvv. The ObjectName is yyy “

「長期にわたるスローダウン状態」を検出すると、メッセージ出力とともに、該当プロセスグループのスタックトレースを採取します。この AP ログに出力されるスタックトレースを参照することで、スローダウンの原因を調査することができます。

詳しくは「運用編（TPモニタの運用操作） 2.40 運用アシスタントに関する設定」を参照してください。
スローダウン検出後は、ジャーナル、オペレーションジャーナル、イベントジャーナルなどの統計データから原因を絞り込むことができます。詳しくは オペレーション遅延への対応を参照してください。

2.2. モジュール別採取情報

各モジュール単位採取すべき情報について説明します。

2.2.1. HTTPサーバ

${ INSTANCE_ROOT}/logs/WebServer ディレクトリのファイル全て
（access.log/error.log 等）

${ INSTANCE_ROOT}/config/WebServer ディレクトリのファイル全て
（httpd.conf/ssl.conf 等）

2.2.2. Webコンテナ

${ INSTANCE_ROOT}/logs ディレクトリのファイル全て

2.2.3. EJBコンテナ

- Standard-J Edition の場合

  ${ INSTANCE_ROOT}/logs ディレクトリのファイル全て

- Standard/Enterprise Edition の場合

  ${ INSTANCE_ROOT}/logs ディレクトリのファイル全てと、${ INSTANCE_ROOT}/logs/tpsystem 配下の全ファイル

  Standard/Enterprise Edition の EJB コンテナの詳細ログを採取する場合は、プロセスグループのトレースレベルを 7 に設定してください。

2.2.4. JNDIサービス

${ INSTANCE_ROOT}/log/webotx_jndisp.log

2.2.5. JMS

${ INSTANCE_ROOT}/wojms/instances/wojmsbroker/props/config.properties
JMS サーバに関する設定情報が格納されるファイルです。

${ INSTANCE_ROOT}/logs/wojms/wojmsserver.log
JMS サーバに関する動作がロギングされるファイルです。ファイルの切替が発生すると、wojmsserver.log,wojmsserver_1.log,…,wojmsserver_9.log という名前（デフォルトで最大 10 個）のファイルが作成されます。

${ INSTANCE_ROOT}/logs/wojms/wojmsadmin.log
JMS サーバに対する運用管理操作の履歴がロギングされるファイルです。ファイルの切替が発生すると、wojmsadmin.log,wojmsadmin_1.log,…,wojmsadmin_9.log という名前（デフォルトで最大 10 個）のフ

ァイルが作成されます。

${ INSTANCE_ROOT}/logs/wojms/wojmspacket.log
JMS サーバに対して送受信されるパケットの情報がロギングされるファイルです。ファイルの切替が発生すると、wojmspacket.log,wojmspacket_1.log,…,wojmspacket_9.log という名前（デフォルトで最大 10 個）のファイルが作成されます。

${ INSTANCE_ROOT}/logs/wojms/wojmsmessage.log
JMS メッセージのライフサイクル情報がロギングされるファイルです。ファイルの切替が発生すると、wojmsmessage.log,wojmsmessage_1.log,…,wojmsmessage_9.log という名前（デフォルトで最大 10 個）のファイルが作成されます。

${ INSTANCE_ROOT}/logs/wojms/wojmserror.log
JMS サーバに関するエラーがロギングされるファイルです。ファイルの切替が発生すると、wojmserror.log,wojmserror_1.log,…,wojmserror_9.log という名前（デフォルトで最大 10 個）のファイルが作成されます。

${ INSTANCE_ROOT}/logs/wojms/std.log
JMS サーバ起動時のエラーや、スレッドのダンプの取得結果がロギングされるファイルです。JMS サーバが起動するタイミングでファイルを切り替え、直前に作成されたファイルのみ std.log.bak という名前で残します。

${ INSTANCE_ROOT}/logs/wojms/wojms.log
JMS クライアントに関する動作がロギングされるファイルです。ファイルの切替が発生すると、wojms.log,wojms.log.1,wojms.log.2 という名前（デフォルトで最大 3 個）のファイルが作成されます。

${ INSTANCE_ROOT}/logs/wojms/wojmssv.pid
JMS サーバ起動時の java プロセスのプロセス ID が出力されるファイルです。

2.2.6. Transactionサービス

${ INSTANCE_ROOT}/config/TS 配下の全ファイル
このディレクトリ配下に格納されるのは Transaction サービス内部で使用する設定情報ファイル、および実行中トランザクションの情報が格納されたファイルなどです。

${ INSTANCE_ROOT}/logs/TS 配下の全ファイル
「*.trc」という名前のファイルが格納されますが、これらは Transaction サービスの動作に関する情報を採取したトレースファイルです。ファイルのサイズや情報採取レベルについては otxadmin コマンドにより設定することが可能です。
採取するレベルは 0(採取しない)～5(デバッグレベル)まで 6 段階あります。レベルを上げると情報量は増えますがその分トランザクションの処理速度が遅くなります。通常はレベルを低い設定にしておき、障害が発生した際にレベルを上げて原因を特定するのが通常の運用です。

またこのファイルの内容を参照するためのツールを Transaction サービスで提供しています。
Windows 版では GUI ツールである「トレースビューア」、UNIX、Linux 版では「trcview」l コマンドです。
トレースビューアは<WebOTX インストールディレクトリ>¥TS¥bin の配下にある trcview.exe です。また trcview コマンドは<WebOTX インストールディレクトリ>/TS/sbin の配下にあります。トレースビューアの使い方はツールのメニューからヘルプファイルを参照してください。また trcview コマンドについてはリファレンス編の中にある Transaction サービスの項を参照してください。

2.2.7. TPモニタ・マネージャ

${ INSTANCE_ROOT}/logs/tpsystem/webotx_tpmmgr.log
TP モニタ・マネージャに関する動作がロギングされるファイルです。ファイルの切り替えが発生すると、webotx_tpmmgr.log, webotx_tpmmgr.log.1, webotx_tpmmgr.log.2 という名前（デフォルトで最大 3 個）のファイルが作成されます。

${ INSTANCE_ROOT}/config/tpsystem の次のファイル
- history.act,history.sav : サーバアプリケーションの起動、停止履歴
- sysmsg.trc,sysmsg.sav : AP サーバへの接続履歴

.sav ファイルはそれぞれのバックアップファイルです。50000 行を超えるか tpsystem 起動時にファイルが切り替わりますので、障害発生時はサービス再起動前に採取してください。

${ INSTANCE_ROOT}/logs/tpsystem 配下の全ファイル
アプリケーショングループとプロセスグループ別にサーバアプリケーションのログファイルが出力されます。

2.2.8. OLF/TPアダプタ

オプション製品 – OLF/TP アダプタ –〈WebOTX OLF/TP アダプタ 運用ガイド〉を参照してください。

2.3. 予防措置

以下にあげる措置を講じておくと、障害予防または障害発生時の早期解決に役立ちます。

- ネットワークキャプチャソフト
  Windows では標準ではネットワークキャプチャソフトがありませんが、通信障害時の調査に苦労することがあります。可能であれば、事前にネットワークキャプチャソフトをインストールしておくことをお勧めします。
- カーネルパラメータ設定
  カーネルパラメータをぎりぎりに設定すると、新たにアプリケーションを追加したときなどにマシン再起動が必要になってなってしまいます。カーネルパラメータは、ある程度余裕をもって設定するようにしてください。
- マルチスレッド評価
  処理量増加などにより多重度を増やす時に、通常はスレッド数を増やしますが、1 を 2 に変更するのは十分な事前評価を要します(AP がマルチスレッド対応していない場合は正常に動作しない)。プロセス数を増やすことになると、メモリへの影響があります。事前に 2 スレッド以上で評価/運用されていると、スレッド数の変更が容易になります。
- クライアントアプリケーションでのエラー処理
  クライアントアプリケーションで NO_RESPONSE などのエラー発生時、通信相手が不明だと調査が難航することがあります。通信相手(名前サーバ、OTX、JNDI、…)がわかるように、クライアントアプリケーションではこまめにエラー処理するようにしておくと障害解析性が向上します。
- シスログ設定
  Solaris ではデフォルトのシスログ設定のままでは、Warning などのメッセージがフィルタリングされてしまいます。WebOTX では以下の設定でシスログを出力しています。
  機能識別子(facility):user
  レベル:err、warning、info
  障害解析を容易にするため、あらかじめこれらをファイルに出力するように設定しておくことをお勧めします。設定方法については OS 側のマニュアル man –s 4 syslog.conf をご参照下さい。
  Linux 環境においてはシスログ出力を有効にするためには syslogd に' –r' オプションをつけてください。' –r' オプションがないと出力されません。
- 正常運用時の情報採取
  ある程度順調に動作していたシステムで普段より動作が遅くなったという場合、比較対象となる普段のデータがないと調査に苦労することがあります。キュー滞留数、ジャーナル、オペレーションジャーナルのデータを普段から採取しておくと、スローダウン障害発生時に解析が容易になります。
- JavaVM オプション
  HP-UX の Java プロセスは標準でプロセス独自の時計を持っているためにプロセス間で時刻がずれてトレースの照合が困難になることがあります。これを OFF に設定する(プロセスグループ-JavaVM オプション-その他引数に-XX:+UseGetTimeOfDay を指定)と障害解析が容易になります。
- ワトソン博士の設定
  Windows では例外発生時の OS 側挙動を設定できるようになっています。
  よく、例外発生のダイアログが画面に出力されることがありますが、このダイアログは例外を起こしたプロセスの処理の延長で出しているので、OK を押してダイアログを閉じるまでは例外プロセスはまだ終了していないことになります。終了すれば WebOTX 側の機能で再起動しますが、無人運転状態では OK を押す人がいないためにプロセスがいつまでも終了直前のまま残ってしまい、サービスが継続できない状況となってしまいます。そこで、例外発生時の OS 側挙動としてはダイアログを出さずに自動でワトソンログに出力することを推奨します。
  但し、この設定はマシン一意なので他プロダクトの都合や運用手順と調整する必要があるかもしれません。

2.4. 開発元へのお問い合わせ

開発元へお問い合わせの際は、以下の情報をお知らせください。

WebOTX に関する情報

- WebOTX の Edition
- WebOTX のバージョン
- WebOTX のパッチ適用状況

環境に関する情報

- OS 情報
- JDK のバージョン
- 最近変更した内容

システム構成に関する情報

- 使用言語
- ステートレス/ステートフル

障害に関する情報

- 本番環境/評価環境
- 再現性の有無
- 障害内容とログ

3.障害対策

障害が発生した場合の対策について説明します。

3.1. プロセス監視

外部ツールなどを用いて、WebOTX のプロセス監視を行ないたい場合、監視を行なうべきプロセスおよび復旧方法について説明します。実際にプロセス監視を行なう場合は、以下を参考に必要に応じて監視プロセスの選択を行なってください。

なお、WebOTX上で動作するプロセス一覧については、「運用編(ドメインの運用)」の「6.システムに関する設定」を参照ください。

3.1.1. WebOTXエージェント

監視プロセス

java -server -Dwebotx.funcid=agent -Ddomain.name=domain1 -Xms64m -Xmx512m -XX:NewRatio=2 -Dsun.io.useCanonCaches=false ...

監視を行なう場合の注意点

java プロセスであるため引数まで識別してプロセスを特定する必要があります。エージェントの Java プロセスは、-Dwebotx.funcid=agent のシステムプロパティにより識別可能です。ドメインは、-Ddomain.name システムプロパティにより識別可能です。

異常検出時の復旧方法

エージェントプロセスの監視で異常を検出した場合は該当ドメインの再起動が必要です。

(停止) otxadmin> stop-domain domain1

(起動) otxadmin> start-domain domain1

3.1.2. TPシステム

監視プロセス

iiop リスナ

iioplsn PLUNAME (5151) ...

TP モニタ

tpmonitor -n MySystem

TP モニタプロセスは、tpmonitor が親プロセス、tpmMain が子プロセスになります。tpmMain が異常終了すると、tpmonitor も終了します。従って、tpmonitor を監視すれば、両方の監視ができます。ただし、tpmonitor だけが終了した場合、実質的な業務への影響はほとんどありません。

監視を行なう場合の注意点

iiop リスナ:PLUNAME で示されている番号が、IIOP リスナのポート番号(tpsystem.IIOPListener.listenerPortNumber)です。

TP モニタ: -n の次が WebOTX システム名となっています。

異常検出時の復旧方法

iiop リスナプロセスの監視で異常を検出した場合は該当ドメインの TP システムの再起動が必要です。

(停止) otxadmin> stop-system

(起動) otxadmin> start-system

TP モニタプロセスの監視で異常を検出した場合は該当マシンの再起動が必要です。クラスタ構成の場合は縮退もしくはフェイルオーバさせてください。

(停止) otxadmin> stop-system

(起動) otxadmin> start-system

3.1.3. ObjectBrokerサービス

監視プロセス

namesv

/usr/lib/ObjectSpinner/bin/namesv -WOdomain /opt/WebOTX/domains/domain1

cnamesv

/usr/lib/ObjectSpinner/bin/cnamesv -WOdomain /opt/WebOTX/domains/domain1

oad

/usr/lib/ObjectSpinner/bin/oad -WOdomain /opt/WebOTX/domains/domain1

oadj

java -Dwebotx.funcid=oadj -Ddomain.name=domain1 -classpath /usr/lib/ObjectSpinner/lib/ospiorb50…

ospprxy

/usr/lib/ObjectSpinner/bin/ospprxy …

監視を行なう場合の注意点

namesv, cnamesv, oad, oadj の場合、domain1 の部分がドメイン名となります。

異常検出時の復旧方法

namesv、oad

WebOTX の全ての Edition 製品にインストールされます。

異常を検出した場合は該当ドメインの再起動が必要です。

(停止) otxadmin stop-domain domain1

(起動) otxadmin start-domain domain1

oadj

WebOTX の全ての Edition 製品にインストールされます。

異常を検出した場合は該当ドメインの oadj の再起動が必要です。

(停止) otxadmin stop-oadj

(起動) otxadmin start-oadj

cnamesv

Enterprise Edition で、選択した場合にだけインストールされます。

異常を検出した場合は該当ドメインの cnamesv の再起動が必要です。

(停止) invoke server.objectbrokerservice.cnamesv.stop

(起動) invoke server.objectbrokerservice.cnamesv.start

それでも復旧しない場合は、該当ドメインの再起動が必要です。

(停止) otxadmin stop-domain domain1

(起動) otxadmin start-domain domain1

ospprxy

Enterprise Edition で、選択した場合にだけインストールされます。

異常を検出した場合は ospprxy の再起動が必要です。

(停止) stop-ospprxy

(起動) start-ospprxy

それでも復旧しない場合は、該当ドメインの再起動が必要です。

(停止) otxadmin stop-domain domain1

(起動) otxadmin start-domain domain1

3.1.4. JMSサービス

監視プロセス

JMS サーバ

java -server -Dwebotx.funcid=jms -Ddomain.name=domain1 -cp ....

監視を行なう場合の注意点

java プロセスであるため引数まで識別してプロセスを特定する必要があります。JMS サーバの Java プロセスは、-Dwebotx.funcid=jms のシステムプロパティにより識別可能です。ドメインは、-Ddomain.name システムプロパティにより識別可能です。

異常検出時の復旧方法

異常を検出した場合は該当ドメインの JMS サーバの再起動が必要です。

(停止) otxadmin stop-jms

(起動) otxadmin start-jms

それでも復旧しない場合は、該当ドメインの再起動が必要です。

(停止) otxadmin stop-domain domain1

(起動) otxadmin start-domain domain1

3.1.5. Transactionサービス

監視プロセス

RCS(C++)
rcssv domain1C …

監視を行なう場合の注意点

RCS(C++)
domain1C の domain1 の部分がドメイン名となります。

異常検出時の復旧方法

異常を検出した場合はトランザクションサービスの再起動が必要です。
(停止) otxadmin stop-transaction-service

(起動) otxadmin start-transaction-service

それでも復旧しない場合は該当ドメイン再起動が必要です。

(停止) otxadmin stop-domain domain1

(起動) otxadmin start-domain domain1

3.1.6. Webサーバ(Apache)

インストール時に、Web サーバ(Apache)を選択した場合に有効です。Java の内蔵 Web サーバを利用する場合には、本プロセスを監視する必要はありません。

監視プロセス

利用する Apache のバージョンにより、監視するプロセスが異なります。

(Apache 1.3.x 利用時)
(UNIX) /opt/WebOTX/WebServer/bin/httpd ……
(Windows) <*INSTALLDIR*>¥WebServer¥apache.exe

(Apache 2.0.x 利用時)
(UNIX) /opt/WebOTX/WebServer2/bin/httpd ……
(Windows) <*INSTALLDIR*>¥WebServer2¥bin¥apache.exe

監視を行なう場合の注意点

Apache は、UNIX の場合、1 つの親プロセスと複数の子プロセスが動作し、その子プロセスはアクセス数に応じて自動的に増減しますので監視は親プロセスのみを対象にしてください。Windows の場合、1つの親プロセスと1つ(複数スレッド)の子プロセスが動作しますので、両方のプロセスを監視対象にしてください。

異常検出時の復旧方法

異常を検出した場合は、該当ドメインの Web サーバ(Apache)の再起動が必要です。

(停止) otxadmin invoke server.internal-lifecycle-module.WebServerService.stop

(起動) otxadmin invoke server.internal-lifecycle-module.WebServerService.start

それでも復旧しない場合は、該当ドメインの再起動が必要です。

(停止) otxadmin stop-domain domain1

(起動) otxadmin start-domain domain1

3.2. ログ監視

外部ツールなどを用いて、WebOTX のログ監視を行ないたい場合、監視を行なうべきログファイルおよび監視メッセージについて説明します。実際にログ監視を行なう場合は、以下を参考に必要に応じて監視ログの選択を行なってください。

3.2.1. イベントログ

説明

Windows OS でイベントログ(アプリケーションログ)に出力する WebOTX が出力する障害メッセージを監視させることが可能です。

監視メッセージ

以下のイベントソースが WebOTX で出力するイベントです。

イベントソース名	説明
WebOTX AS	WebOTX エージェントが出力するイベントログです。致命的なメッセージを監視するには「エラー」レベルを監視対象にします。
WebOTXAgentService	「WebOTX Agent Service」が出力するイベントログです。Windows のサービス経由で WebOTX の起動・停止を行なったときのメッセージが出力されます。致命的なメッセージを監視するには「エラー」レベルを監視対象にします。
WebOTX_ASMonitor *systemname*	Standard/Enterprise Edition における「TP モニタ」サービスが出力するイベントログです。*systemname* には該当ドメインで設定したシステム名が入ります。監視の詳細は 3.2.4 章を参照してください。
WebOTX_ASMonitor	Standard/Enterprise Edition における TP モニタ制御サービス（TPA サーバ:WebOTX TPBASEadm サービス）が出力するイベントログです。監視の詳細は 3.2.4 章 「TPS12:TPA サーバ（運用管理）が出力するメッセージ」を参照してください。
WebOTX_TS	旧バージョンの WebOTX Transaction サービスが出力するイベントログです。Transaction サービスの旧互換ライブラリを使用している場合のみ出力します。致命的なメッセージを監視するには「エラー」レベルを監視対象にします。

3.2.2. シスログ

説明

UNIX OS でシスログに出力する WebOTX が出力する障害メッセージを監視させることが可能です。

監視メッセージ

以下のイベントソースが WebOTX で出力するイベントです。

ログ識別名	説明
WebOTX AS	WebOTX 内部で共有メモリやセマフォの作成、取得に失敗した場合に出力するイベントログです。また、Transaction サービスが出力するイベントログです。致命的なメッセージを監視するには「Error」レベル以上を監視対象にします。
WebOTX_Agent	WebOTX エージェントが出力するイベントログです。致命的なメッセージを監視するには「Error」レベル以上を監視対象にします。

OTXM:*systemname*	Standard/Enterprise Edition における「TP モニタ」サービスが出力するイベントログです。*systemname* には該当ドメインで設定したシステム名が入ります。監視の詳細は 3.2.4 章を参照してください。
WebOTX_TS	旧バージョンの WebOTX Transaction サービスが出力するイベントログです。Transaction サービスの旧互換ライブラリを使用している場合のみ出力します。致命的なメッセージを監視するには「Error」レベル以上を監視対象にします。

3.2.3. WebOTX独自ログ

説明

WebOTX で独自に出力するログファイルを監視させることが可能です。

監視メッセージ

監視対象にするファイルについて以下に説明します。

ファイル名	説明
${INSTANCE_ROOT}/logs WebOTX_Agent.log	WebOTX ドメインのエージェント内の JVM で出力される全てのメッセージを統合したログでです。致命的なメッセージを監視するには「ERROR」レベルを監視対象にします。
${INSTANCE_ROOT}/logs/WebServer error_log	WebOTX WebServer のエラーログです。致命的なメッセージを監視するにはログの先頭に「error」または「emerg」が出力されたメッセージを監視対象にします。
${INSTANCE_ROOT}/logs/ObjectBroker oad.log	Object Broker の oad が出力するログファイルです。致命的なメッセージを監視するには、「ERROR: Startup of oad faild.」を含むメッセージを監視対象にします。
${INSTANCE_ROOT}/logs/ObjectBroker namesv.log	Object Broker の名前サーバが出力するログファイルです。致命的なメッセージを監視するには、「ERROR: Startup of namesv faild.」を含むメッセージを監視対象にします。
${INSTANCE_ROOT}/logs/ObjectBroker cnamesv.log	Object Broker のキャッシュ名前サーバが出力するログファイルです。致命的なメッセージを監視するには、「ERROR: Startup of cnamesv faild.」を含むメッセージを監視対象にします。

3.2.4. 監視対象メッセージ(TPSメッセージ)

説明

イベントログ・シスログで監視すべき TPS メッセージは基本的に ERROR レベルになります。ただし、ERROR レベルであっても条件によっては正常時に出力されるメッセージや、INFO レベルであっても監視対象にすべきメッセージが存在します。これらのメッセージ監視について説明します。

監視メッセージ

監視対象にするメッセージ ID とエラーレベル、説明と注意事項について以下に説明します。メッセージ内容や処置についてはオペレータメッセージ編を参照してください。

TPS01:内部テーブルアクセスが出力するメッセージ

監視メッセージは特にありません。

TPS04:システム TPP が出力するメッセージ

以下のメッセージ ID を監視対象としてください。TPS04-00703 と TPS04-00713 以外の ERROR メッセージを監視する必要があります。

メッセージ ID	エラーレベル	説明・注意事項
TPS04-00102	ERROR	システム TPP の初期処理でエラーが発生したので、システム TPP は終了しました。エラーには以下の場合があります。 1. キュー制御初期化時のエラー 2. テーブル参照の失敗 3. 環境変数の不正
TPS04-00103	ERROR	キューの送受信で永久障害が発生したので、システム TPP は終了しました。
TPS04-00105	ERROR	TPBASE 内部テーブルのロック/アンロックの異常により、システム TPP は終了しました。
TPS04-00203	ERROR	WebOTX 内部で使用するテーブルが正しくありません。
TPS04-00204	ERROR	カタログディレクトリ下の tpsmsg ディレクトリへの移動に失敗しました。ERRNO は、内部で発行しているシステムの chdir 関数の errno です。
TPS04-00406	ERROR	端末情報の取得に失敗しました。
TPS04-00703	ERROR	共有作業域確保/ 解放関数でエラーが発生しました。
TPS04-00807	ERROR	アプリケーション初期プロセスが異常終了しました。
TPS04-00813	ERROR	アプリケーション初期プロセスを起動できません。
TPS04-00816	ERROR	アプリケーション初期プロセスからのレスポンスを、待ち時間を経過した後に受信しました。トランザクションの実行結果に関係なく電文を破棄しました。負荷が高い状態で運用されていることが考えられます。
TPS04-00903	ERROR	オペレータメッセージの送信処理でエラーが発生しました。
TPS04-00905	ERROR	オペレータメッセージの送信処理でエラーが発生しました。
TPS04-01002	ERROR	アプリケーショングループの起動に失敗しました。

TPS04-01003	ERROR	アプリケーショングループの停止に失敗しました。
TPS04-01011	ERROR	内部テーブルが正しくありません。
TPS04-01012	ERROR	VD サーバから不正な電文を受信しました。
TPS04-01013	ERROR	VD サーバに対する電文の送信でエラーが発生しました。
TPS04-01201	ERROR	キュー制御関数でエラーが発生しました。一時障害なら処理を続行します。永久障害ならシステム TPP は終了します。
TPS04-01202	ERROR	メモリプール関数でエラーが発生しました。一時障害なら処理を続行します。永久障害ならシステム TPP は終了します。
TPS04-01203	ERROR	キュー制御関数でエラーが発生しました。一時障害なら処理を続行します。永久障害ならシステム TPP は終了します。
TPS04-01301	ERROR	メモリ不足のため作業領域が確保できません。
TPS04-01302	ERROR	割り込みが発生しました。SIGID はシグナル種別です。**SIGID=15 であれば、モニタ正常終了時でも出力されることがあります。**
TPS04-01404	ERROR	VD の起動に失敗しました。
TPS04-01405	ERROR	VD の停止に失敗しました。
TPS04-01503	ERROR	active ディレクトリへのアクセスが失敗しました。

TPS06:キュー制御が出力するメッセージ

監視メッセージは特にありません。

TPS07:通信リスナが出力するメッセージ

以下のメッセージ ID を監視対象としてください。TPS07-00307 以外の ERROR を監視する必要があります。

メッセージ ID	エラーレベル	説明・注意事項
TPS07-00106	ERROR	リスナでキュー制御関数が失敗しました。エラーの内容は ERRNO に依存します。
TPS07-00312	ERROR	内部テーブルにアタッチできません。
TPS07-00313	ERROR	テーブルのロック／アンロックでエラーが発生しました。
TPS07-00401	ERROR	リスナが起動後、着呼待ちを行うための通信手順を行う際に致命的エラーがメッセージ内に表示される通信 API 関数で発生しました。
TPS07-00402	ERROR	リスナ起動に必須である関数（メッセージ内に表示）でエラーが発生したため、リスナが起動できません。
TPS07-00403	ERROR	ッセージ内に表示されるキュー制御関数で致命的エラーが発生しました。異常の内容は ERRNO に依存します。
TPS07-00602	ERROR	TPBASE 内部のキュー制御関数でエラーが発生しました。

TPS09:VD サーバが出力するメッセージ

以下のメッセージ ID を監視対象としてください。監視しない ERROR メッセージがあります。

メッセージ ID	エラーレベル	説明・注意事項
TPS09-00101	ERROR	VD サーバの初期化処理に失敗しました。
TPS09-00106	WARNING	VD サーバが VD 解放処理に失敗しました。VD を閉塞します。
TPS09-00107	WARNING	VD サーバが VD 接続処理に失敗しました。VD を閉塞します。
TPS09-00114	ERROR	電文の再送に失敗しました。
TPS09-00115	WARNING	トランザクション型 VD で起動先オペレーションのレスポンス待ち時間をオーバーしました。
TPS09-00202	WARNING	VD サーバが VD ごとの制御テーブルを検索できませんでした。
TPS09-00203	WARNING	VD サーバがクライアントごとの制御テーブルを検索できませんでした。
TPS09-00204	WARNING	VD サーバがオペレーションごとの制御テーブルを検索できませんでした。
TPS09-00205	WARNING	VD サーバが接続中のクライアントごとの制御テーブルを検索できませんでした。
TPS09-00206	WARNING	VD サーバが接続中の VD ごとの制御テーブルを検索できませんでした。
TPS09-00309	WARNING	メモリの確保に失敗しました。
TPS09-00401	ERROR	VD サーバがメモリプールの初期化をできませんでした。
TPS09-00402	ERROR	VD サーバがメモリプールの生成をできませんでした。
TPS09-00403	ERROR	VD サーバがメモリプールの削除をできませんでした。
TPS09-00404	ERROR	VD サーバが端末送受信用メモリプール ID の取得に失敗しました。
TPS09-00406	ERROR	電文送信用に確保したメモリブロックの解放に失敗しました。
TPS09-00501	ERROR	VD サーバがキュー制御の初期化をできませんでした。
TPS09-00502	ERROR	VD サーバがキューを生成できませんでした。
TPS09-00503	ERROR	VD サーバがキューを削除できませんでした。
TPS09-00504	ERROR	VD サーバがキュー制御のリセットをできませんでした。
TPS09-00506	WARNING	VD サーバがキューをデキューイング禁止にできませんでした。
TPS09-00507	WARNING	VD サーバがキューをデキューイング禁止解除にできませんでした。
TPS09-00509	WARNING	VD サーバがトランザクション型 VD の電文を送信できませんでした。この VD を閉塞します。
TPS09-00511	WARNING	VD サーバが端末型 VD の電文を送信できませんでした。この VD を閉塞します。
TPS09-00512	WARNING	正常に送信完了した電文をキューから削除できませんでした。この VD を閉塞します。
TPS09-00513	WARNING	VD サーバがキューをキューイング禁止にできませんでした。この VD を閉塞します。
TPS09-00514	WARNING	VD サーバがキューをキューイング禁止解除にできませんでした。この VD を閉塞します。
TPS09-00516	WARNING	VD サーバが滞留電文数を取得するためのキュー情報取得に失敗しまし

		た。この VD を閉塞します。
TPS09-00518	WARNING	VD サーバがキューの電文をパージ(デキューイング)できませんでした。
TPS09-00601	WARNING	リソース不足検出時のリトライ処理のためのタイマ登録に失敗しました。この VD を閉塞します。
TPS09-00602	WARNING	レスポンス時間監視用タイマ登録に失敗しました。
TPS09-00603	WARNING	レスポンス時間監視用タイマキャンセルに失敗しました。
TPS09-01110	WARNING	VD サーバへの VD 滞留電文削除要求に失敗しました。
TPS09-01114	WARNING	VD 作成に失敗しました。
TPS09-01119	WARNING	VD キューの滞留電文数の取得に失敗しました。
TPS09-01124	WARNING	処理の途中でクライアントが切断されました。
TPS09-01201	WARNING	ファイル型 VD のファイルオープンでエラーが発生しました。
TPS09-01202	WARNING	ファイル型 VD のファイルリードでエラーが発生しました。
TPS09-01203	WARNING	ファイル型 VD のファイルライトでエラーが発生しました。
TPS09-01204	WARNING	ファイル型 VD のメッセージ登録に失敗しました。

TPS10:tplib(サーバ AP ライブラリ)が出力するメッセージ

以下のメッセージ ID を必須の監視対象としてください。

メッセージ ID	エラーレベル	説明・注意事項
TPS10-00301	ERROR	メモリプール ID が取得できません。
TPS10-00401	ERROR	内部で利用される共有メモリが不足しています。
TPS10-00501	ERROR	メモリプールの初期化ができません。
TPS10-00601	ERROR	メモリブロックの解放ができません。
TPS10-00701	ERROR	メモリプールの作成ができません。
TPS10-00801	ERROR	内部で利用されるキュー制御関数の初期化ができません。
TPS10-00901	ERROR	内部で利用されるキューの生成ができません。
TPS10-01001	ERROR	内部で利用されるキュー制御関数でエラーが発生しました。
TPS10-01101	ERROR	内部で利用されるキュー制御関数でエラーが発生しました。
TPS10-01201	ERROR	内部で利用されるキュー制御関数でエラーが発生しました。
TPS10-01301	ERROR	内部で利用されるメモリブロックの確保ができませんでした。
TPS10-01401	ERROR	内部で利用される共有メモリが不足しています。
TPS10-02501	ERROR	指定されたプロセスグループは存在しない。
TPS10-02601	ERROR	なんらかの原因で内部で利用されるテーブルが破壊されています。
TPS10-02701	ERROR	なんらかの原因で内部で利用されるテーブルが破壊されています。
TPS10-02901	ERROR	メモリ不足です。
TPS10-03001	ERROR	active ディレクトリに書き込み権がありません。
TPS10-04501	ERROR	なんらかの原因で内部で利用されるテーブルが破壊されています。
TPS10-11201	ERROR	受信スレッドの停止状態を検出しました。

TPS10-11301	ERROR	送信スレッドの停止状態を検出しました。
TPS10-11701	ERROR	実行スレッドが全て停止しました。
TPS10-13501	ERROR	WebOTX の制御外でスレッドが消滅しているのを検出した。
TPS10-14201	ERROR	内部で利用されるキュー制御関数でエラーが発生しました。

以下のメッセージ ID は利用者側の判断で監視対象としてください。

メッセージ ID	エラーレベル	説明・注意事項
TPS10-01901	ERROR	メモリ不足です
TPS10-02101	ERROR	メモリ不足です
TPS10-02201	ERROR	メモリ不足です
TPS10-02301	ERROR	メモリ不足です
TPS10-03101	ERROR	停止中のアプリケーショングループに対して呼び出しが発生しました。
TPS10-03201	ERROR	閉塞中のオペレーションに対して呼び出しが発生しました。
TPS10-03301	ERROR	サーバコンポーネント(dll ディレクトリの.so、.sl、.dll)の形式不正。
TPS10-03501	ERROR	データベースの接続ができませんでした。
TPS10-03601	ERROR	データベースの切り離しに失敗しました。
TPS10-03701	ERROR	オペレーションの再実行回数が設定値になりました。
TPS10-04401	ERROR	オペレーション実行中に例外が発生しました。コードに続いてアプリケーショングループ名、プロセスグループ名、プロセス ID を表示しています。
TPS10-04901	ERROR	なんらかの原因で内部テーブルが破壊されています。
TPS10-05101	ERROR	ディスク容量不足で、ファイル型 VD への書き込みエラーが発生しました。
TPS10-05301	ERROR	ORACLE のエラーが検出されました。
TPS10-11001	ERROR	スレッド情報の初期化に失敗しました。
TPS10-11101	ERROR	スレッド生成後の初期化処理にてタイムアウトが発生しました。
TPS10-12001	ERROR	内部エラー。
TPS10-12201	ERROR	内部エラー。
TPS10-12301	ERROR	内部エラー。
TPS10-13001	ERROR	TPSAbort(0)（スレッドの終了）が呼び出されました。ユーザ実装部で TPSAbort(0)を呼んだか、もしくは、Java で作られた常駐オブジェクトのコールバックがランタイム例外を発生したためにそれを検出したオブジェクトマネージャが TPSAbort(0)を呼び出しました。
TPS10-13101	ERROR	TPSAbort(1)（スレッドのリスタート）が呼び出されました。
TPS10-13201	ERROR	TPSAbort(-1)（プロセスの終了）が呼び出されました。ユーザ実装部で TPSAbort(-1)を呼んだか、もしくは、Java で作られたユーザ実装部がランタイム例外を発生したためにそれを検出したオブジェクトマネージャが TPSAbort(-1)を呼び出しました。
TPS10-13301	ERROR	オペレーションの実行時間上限を超えたため処理を打ち切りました。

TPS10-14001	ERROR	メモリ不足です。
TPS10-14101	ERROR	メモリ不足です。
TPS10-16001	ERROR	DB 連携機能を使用したプロセスで例外が発生したためプロセスを強制終了しました。
TPS10-16101	ERROR	DB 連携機能を使用したプロセスで実行時間超過が発生したためプロセスを強制終了しました。

TPS11:ジャーナルが出力するメッセージ

監視メッセージは特にありません。

TPS12:TPA サーバ(運用管理)が出力するメッセージ

以下のメッセージ ID を監視対象としてください。

メッセージ ID	エラーレベル	説明・注意事項
TPS12-00101	ERROR	ホスト名が hosts ファイル内に定義されていません。
TPS12-00103	ERROR	socket システムコールでエラーが発生しました。
TPS12-00104	ERROR	connect システムコールでエラーが発生しました。
TPS12-00112	ERROR	**Windows のみ** サービス制御のためのハンドル取得に失敗しました。
TPS12-00113	ERROR	**Windows のみ** ステータス取得に失敗しました。
TPS12-00115	ERROR	**Windows のみ** bind システムコールでエラーが発生しました。
TPS12-00116	ERROR	**Windows のみ** listen システムコールでエラーが発生しました。
TPS12-00117	ERROR	**Windows のみ** accept システムコールでエラーが発生しました。
TPS12-00119	ERROR	getaddrinfo システムコールでエラーが発生しました。
TPS12-00120	ERROR	ソケットの作成に失敗しました。
TPS12-00121	ERROR	select システムコールでエラーが発生しました。
TPS12-01310	ERROR	**Windows のみ** WSAStartup の実行に失敗しました。
TPS12-01311	ERROR	**Windows のみ** OpenSCManager の実行に失敗しました。
TPS12-01312	ERROR	**Windows のみ** TPBASEadm の起動に失敗しました。
TPS12-01313	ERROR	**Windows のみ** 起動待ち時間内に TPBASEadm の起動が完了しませんでした。

TPS12-54101	ERROR	**Windows のみ** サービス WebOTX AS TPBASEadm 起動時に、エラー番号の要因により、レジストリ(SOFTWARE¥NEC¥WebOTX¥TPBASE)のオープン関数 RegOpenKeyEx()に失敗しました。
TPS12-54102	ERROR	**Windows のみ** サービス WebOTX AS TPBASEadm 起動時に、エラー番号の要因により、レジストリキー値の取得関数 RegQueryValueEx()に失敗しました。
TPS12-54112	ERROR	**Windows のみ** サービス WebOTX AS TPBASEadm 起動時に、エラー番号の要因により、StartServiceCtrlDispatcher()の実行に失敗しました。
TPS12-54132	ERROR	**Windows のみ** サービス WebOTX AS TPBASEadm 起動時に、エラー番号の要因により、RegisterServiceCtrlHandler()の実行に失敗しました。
TPS12-54133	ERROR	**Windows のみ** サービス WebOTX AS TPBASEadm の起動、または、停止時に、エラー番号の要因により、サービス制御マネージャへの通知関数 ReportStatus()の実行に失敗しました。
TPS12-54134	ERROR	**Windows のみ** サービス WebOTX AS TPBASEadm 起動時に、エラー番号の要因により、CreateProcess()の実行に失敗しました。
TPS12-54239	ERROR	**Windows のみ** サービス WebOTX TPBASEadm 起動後の、サービス監視関数の実行に失敗しました。(リトライ失敗)

TPS13:メッセージ制御が出力するメッセージ

以下のメッセージ ID を監視対象としてください。全 ERROR メッセージを監視対象にする必要があります。

メッセージ ID	エラーレベル	説明・注意事項
TPS13-00102	ERROR	メッセージ制御の初期処理でエラーが発生したため終了しました。以下の場合があります。 ● キュー制御初期化のエラー ● テーブル参照の失敗 ● 環境変数の不正
TPS13-00103	ERROR	キューの送受信で永久障害が発生したので、メッセージ制御は終了しました。
TPS13-00105	ERROR	内部テーブルのロック／アンロックの異常により、メッセージ制御は終了しました。
TPS13-00201	INFO	内部で使用するテーブルが正しくありません。

TPS13-00301	INFO	受信電文に必要な最低限の長さがありません。
TPS13-00302	ERROR	受信電文の要求種別はメッセージ送信 TPP で扱わないものです。
TPS13-00502	ERROR	割り込みが発生しました。

TPS15:TP モニタが出力するメッセージ

以下のメッセージ ID を監視対象としてください。監視しない ERROR メッセージがあります。

メッセージ ID	エラーレベル	説明・注意事項
TPS15-00102	ERROR	プロセスの起動に失敗しました。
TPS15-00106	ERROR	TP システムが起動時情報登録に失敗しました。
TPS15-00602	ERROR	メモリ不足のため、コマンド管理テーブルの作成に失敗しました。
TPS15-00603	ERROR	メモリ不足のため、プロセス管理テーブルの作成に失敗しました。
TPS15-00604	ERROR	共有メモリはすでに存在しています。
TPS15-00605	ERROR	共有メモリの確保に失敗しました。
TPS15-00612	WARNING	メッセージキューが処理待ちのメッセージでいっぱいになりました。
TPS15-00701	ERROR	TPBASE 構成ファイル中の TBLKEY で共有テーブルが作成できませんでした。
TPS15-00702	ERROR	TPBASE 構成ファイル中の TBLKEY で共有テーブルが作成できませんでした。
TPS15-00901	ERROR	構成ファイル(tpbase.cnf)の処理中にエラーが発生しました。
TPS15-01101	ERROR	TP システムの起動に失敗しました。
TPS15-01102	WARNING	TP システムが異常終了しました。
TPS15-01103	ERROR	アプリケーショングループの起動に失敗しました。続く名前はアプリケーショングループ名。
TPS15-01105	ERROR	プロセスグループの起動に失敗しました。CLASS はプロセスグループ名、PED はアプリケーショングループ名。
TPS15-01107	WARNING	TP システムからの停止指示以外の原因でプロセスが終了しました。PID はプロセス ID、class はプロセスグループ名、ped はアプリケーショングループ名。 **リスナ異常終了対策として、TPS15-01107 と class[IIOPLSN_CLS] の and 条件、TPS15-01107 と class[OLFTPLSN_CLS] の and 条件、を監視して下さい。なお、WebOTX を起動したまま OS シャットダウンを行なうと本メッセージがでる場合があることに留意してください。検出時の対処としては、TPS07 で始まるその他のメッセージが出ている場合はその対処に従い、特に出ていない場合は WebOTX 開発元に連絡して下さい。**
TPS15-01313	ERROR	ファイル I/O エラーが発生しました。
TPS15-02002	ERROR	TP モニタプロセス(tpmMain)で例外が発生しました。
TPS15-02003	ERROR	sigaction()コールの失敗により、TP システムの起動に失敗しました。

TPS15-02004	ERROR	fork()コールの失敗により、TP システムの起動に失敗しました。
TPS15-02009	ERROR	TP システム登録情報が不正です。
TPS15-02022	ERROR	レジストリへのアクセスに失敗しました。
TPS15-02023	ERROR	レジストリへのアクセスに失敗しました
TPS15-02026	ERROR	CreateProcess()コールの失敗により、TP モニタの起動に失敗しました。
TPS15-02027	ERROR	sigaction()コールの失敗により、TP モニタの起動に失敗しました。
TPS15-02028	ERROR	execvp()コールの失敗により、TP モニタの起動に失敗しました。
TPS15-02029	ERROR	fork()コールの失敗により、TP モニタの起動に失敗しました。
TPS15-02030	ERROR	waitpid()コールの失敗により、TP モニタの起動に失敗しました。
TPS15-02034	ERROR	sigaction()コールの失敗により、TP モニタの起動に失敗しました。
TPS15-02101	ERROR	TP モニタが予期しないシグナル xxxx を受信し、例外しました。
TPS15-10001	ERROR	キュー管理ロックテーブルの初期化に失敗しました。
TPS15-10002	ERROR	キュー管理テーブルの共有メモリの確保に失敗しました。
TPS15-10003	ERROR	キュー管理テーブルの共有メモリのアタッチに失敗しました。
TPS15-10004	ERROR	メッセージ管理テーブルの共有メモリの確保に失敗しました。
TPS15-10005	ERROR	メッセージ管理テーブルの初期化に失敗しました。
TPS15-10006	ERROR	メモリプールテーブルの共有メモリの確保に失敗しました。
TPS15-10007	ERROR	メモリプールテーブルの初期化に失敗しました。
TPS15-10008	ERROR	メモリプールテーブルの作成に失敗しました。

TPS17:タイマデーモンが出力するメッセージ

監視メッセージは特にありません。

TPS18:データベースインタフェースが出力するメッセージ

監視メッセージは特にありません。

TPS19:VD キュー管理が出力するメッセージ

以下のメッセージ ID を監視対象としてください。全 ERROR メッセージを監視対象にする必要があります。

メッセージ ID	エラーレベル	説明・注意事項
TPS19-00101	ERROR	VD 用データファイルが壊れています。または、未初期化のデータファイルを使用しています。
TPS19-00102	ERROR	VD 用データファイルが壊れています。または、未初期化のデータファイルを使用しています。
TPS19-00103	WARNING	VD 用データファイルの破損を検出しましたが修復しました。
TPS19-00104	WARNING	VD 用データファイルの破損を検出しましたが修復しました。
TPS19-00105	ERROR	VD 用データファイルの設定変更に失敗しました。変更内容は以下の場合があります。

		・filesize：ファイルサイズが現在使用中のサイズより小さく縮小できない。
TPS19-00106	ERROR	VD 用データファイルに空き領域がありません。
TPS19-00107	ERROR	VD 用データファイルが壊れている可能性があります。
TPS19-00201	ERROR	VD 用キューの管理データが壊れています。
TPS19-00202	ERROR	VD 用キューの管理データが壊れています。
TPS19-00203	WARNING	VD 用キューの管理データの破損を検出しましたが修復しました。
TPS19-00204	WARNING	VD 用キューの管理データの破損を検出しましたが修復しました。
TPS19-00207	ERROR	VD 用キューの管理テーブルが存在しません。または、未初期化のデータファイルを使用しています。管理テーブルは以下の場合があります。 ● file management table ：データファイル管理テーブル ● file list table ：ファイルリストテーブル ● queue list table ：キューリストテーブル
TPS19-00208	ERROR	重複するキューを作成しようとしました。
TPS19-00209	ERROR	最大キュー数を越えてキューを作成しようとしました。
TPS19-00210	ERROR	最大ファイル数を越えて VD データファイルを作成しようとしました。
TPS19-00211	ERROR	送信禁止状態のキューにデータを送信しようとしました。VD データファイルが壊れている可能性があります。
TPS19-00212	ERROR	受信禁止状態のキューからデータを受信しようとしました。VD データファイルが壊れている可能性があります。
TPS19-00213	ERROR	優先送信不可状態のキューにデータを優先送信しようとしました。VD データファイルが壊れている可能性があります。
TPS19-00302	ERROR	他プロセスが掛けたロックをアンロックしようとしました。VD データファイルが壊れている可能性があります。
TPS19-00303	ERROR	ロックファイル名称を取得できませんでした。
TPS19-01101	ERROR	WebOTX 内部で使用する SG テーブルが正しくありません。
TPS19-01102	ERROR	最大値を越えた要求です。 ・tpbase.cnf/MAXVDF ：VD データファイル数のオーバー
TPS19-02101	ERROR	OS のライブラリ関数呼び出しでエラーが発生しました。資源不足やシステム異常が発生している可能性があります。
TPS19-02102	ERROR	OS のライブラリ関数呼び出しでエラーが発生しました。資源不足やシステム異常が発生している可能性があります。
TPS19-02103	ERROR	OS のライブラリ関数呼び出しでエラーが発生しました。資源不足やシステム異常が発生している可能性があります。
TPS19-02104	ERROR	OS のライブラリ関数呼び出しでエラーが発生しました。資源不足やシステム異常が発生している可能性があります。
TPS19-02105	ERROR	OS のライブラリ関数呼び出しでエラーが発生しました。資源不足やシステム異常が発生している可能性があります。

TPS20:VD キューツールコマンドが出力するメッセージ

監視メッセージは特にありません。

3.3. Object Broker での例外と対処

Object Broker サービスに関する例外と対処について説明します。

3.3.1. Object Broker Java例外一覧

Object BrokerJava で発生する例外ごとに、マイナーコード別の原因と対策を一覧にしています。

NO_RESPONSE

・マイナーコード 5038 (vmcid: 0x1000　minor code: 942)

原因	指定した初期サービスが見つからない
対策	指定した初期サービスが正常に起動しているかを確認してください。

・マイナーコード 5130 (vmcid: 0x1000　minor code: 1034)

原因	リクエスト呼び出しにおいてタイムアウトが発生
対策	サーバの処理に時間がかかっていないかを確認してください。CORBA リクエスト呼び出しのタイムアウト値は、デフォルトで 30 秒となっています。サーバの処理に 30 秒以上かかる場合は、統合運用管理ツールで、objectbrokerconfig MO の属性「リクエスト呼び出しのタイムアウト時間」(RequestTimeout)を、適切な値に変更してください。

・マイナーコード 5006 (vmcid: 0x1000　minor code: 910)

原因	メッセージの受信に失敗した
対策	サーバとクライアントのインターフェース(IDL)の定義が合っていない可能性があります。サーバもしくはクライアントのどちらか一方のファイルを更新して、アプリケーションのファイルのバージョンが合っていない状態となっていないかを確認してください。

・マイナーコード 5092 (vmcid: 0x1000　minor code: 996)

原因	名前サーバが見つからない
対策	名前サーバ(namesv)が正常に起動しているかを確認してください。

・マイナーコード 5093 (vmcid: 0x1000　minor code: 997)

原因	IR サーバが見つからない
対策	IR サーバ(irsv)が正常に起動しているかを確認してください。

・マイナーコード 5195 (vmcid: 0x1000　minor code: 1099)

原因	SecurityServiceComponentManager が見つからない
対策	開発部門へお問い合わせください。

・マイナーコード 5100 (vmcid: 0x1000　minor code: 1004)

原因	get_next_response()がタイムアウトした
対策	サーバの処理に時間がかかっていないかを確認してください。CORBA リクエスト呼び出しのタイムアウト値は、デフォルトで 30 秒となっています。サーバの処理に 30 秒以上かかる場合は、統合運用管理ツールで、objectbrokerconfig MO の属性「リクエスト呼び出しのタイムアウト時間」(RequestTimeout)を、適切な値に変更してください。

COMM_FAILURE

・マイナーコード 5007 (vmcid: 0x1000　minor code: 911)

原因	TCP/IP レベルの通信障害発生により、java.io.IOException が検知された
対策	例外メッセージを参照してください。また、通信相手のサーバマシンが起動しているかを確認してください。

・マイナーコード 5027 (vmcid: 0x1000　minor code: 931)

原因	TCP/IP のソケットの開設に失敗した
対策	ソケットがすでにクローズしているためにストリームを取得できませんでした。アプリケーションが終了していないか、WebOTX のシステム/サービスが起動しているか、およびネットワークの状態について確認してください。

・マイナーコード 5037 (vmcid: 0x1000　minor code: 941)

原因	未知のホスト名
対策	ホスト名が正しく指定されているか、DNS などによるホスト名解決で正しく検索可能かを確認してください。

・マイナーコード 5046 (vmcid: 0x1000　minor code: 950)

原因	java.net.SocketException が発生した
対策	名前サービスなどのサービスの初期リファレンスの取得に失敗しました。WebOTX のシステム/サービスが起動しているかを確認してください。

・マイナーコード 5149 (vmcid: 0x1000　minor code: 1053)

原因	これ以上のコネクションを張ることができない
対策	コネクションの数が制限を超えています。コネクションの最大接続数を変更する場合は、統合運用管理ツールで、objectbrokerconfig MO の属性 「コネクションの最大接続数」(MaxConnection)を、適切な値に修正してください。

・マイナーコード 5174 (vmcid: 0x1000　minor code: 1078)

原因	SSL ハンドシェイクが失敗した
対策	指定した証明書の有効期限が切れていないかなど、SSL の環境について確認してください。

・マイナーコード 5224(vmcid: 0x1000　minor code: 1128)

原因	多重化されたオブジェクトリファレンスのコネクション接続に失敗した
対策	コネクション活性化の待ち合わせを行いましたが、コネクション接続に失敗しました。多重化の設定を確認してください。

UNKNOWN

・マイナーコード 5007 (vmcid: 0x1000　minor code: 911)

原因	java.io.IOException が発生した
対策	自動起動サーバ登録情報の読み込みまたは書き込みに失敗しました。以下のディレクトリのファイルの内容を確認してください。 (ドメイン配下で動作する WebOTX システムの場合) Windows　:　(WebOTX インストールディレクトリ)¥domains¥(ドメイン名)¥config¥ObjectBroker¥implrep UNIX　:　/opt/WebOTX/domains/(ドメイン名)/config/ObjectBroker/implrep (ドメイン外で動作する WebOTX システムの場合) Windows　:　(WebOTX インストールディレクトリ)¥Objectbroker¥conf¥implrep UNIX　:　/opt/ObjectSpinner/conf/implrep

・マイナーコード 5038 (vmcid: 0x1000　minor code: 942)

原因	自動起動サーバがみつからない
対策	登録されている自動起動サーバの設定を確認してください。

・マイナーコード 5144 (vmcid: 0x1000　minor code: 1048)

原因	自動起動登録サーバの情報が不正
対策	登録されている自動起動サーバの情報の形式が不正です。自動起動登録設定を確認してください。 以下のディレクトリのファイルの内容を確認してください。 （ドメイン配下で動作する WebOTX システムの場合） Windows : (WebOTX インストールディレクトリ)¥domains¥(ドメイン名)¥config¥ObjectBroker¥implrep UNIX : /opt/WebOTX/domains/(ドメイン名)/config/ObjectBroker/implrep （ドメイン外で動作する WebOTX システムの場合） Windows : (WebOTX インストールディレクトリ)¥Objectbroker¥conf¥implrep UNIX : /opt/ObjectSpinner/conf/implrep

・マイナーコード 5204 (vmcid: 0x1000 minor code: 1108)

原因	不正な Value(オブジェクト直列化仕様違反)が指定され、java.lang.IllegalAccessException が発生した
対策	指定した Value が不正でないか確認してください。

・マイナーコード 1330446337 (vmcid: OMG minor code: 1)

原因	未知のユーザ例外が発生
対策	ユーザコードでなんらかの例外が発生しました。サーバ側のログなどで発生した例外を確認してください。

・マイナーコード 1330446338 (vmcid: OMG minor code: 2)

原因	未知のシステム例外が発生
対策	標準のシステム例外ではない例外が発生しました。サーバ側のログなどで発生した例外を確認してください。

INITIALIZE

・マイナーコード 5007 (vmcid: 0x1000 minor code: 911)

原因	java.io.IOException が発生した
対策	WebOTX のサービス、ドメインが起動しているか、ObjectSpinner サービスが起動しているか、また、ネットワークの状態や、ホスト名が解決できているかどうかなどについて確認してください。

・マイナーコード 5036 (vmcid: 0x1000 minor code: 940)

原因	不正な SSLPort が指定された
対策	SSL を用いた通信で使用するポート番号に、不正な値が指定されていないか確認してください。

・マイナーコード 5037 (vmcid: 0x1000 minor code: 941)

原因	未知のホストが指定された
対策	ホスト名が正しく指定されているか、DNS などによるホスト名解決で正しく検索可能かを確認してください。

・マイナーコード 5044 (vmcid: 0x1000 minor code: 948)

原因	プロパティに不正な値が指定された
対策	Integer を指定すべきプロパティに文字列が指定された、など、不正な値が設定されています。プロパティの設定値を確認してください。

・マイナーコード 5084 (vmcid: 0x1000 minor code: 988)

原因	oad のホスト名が null
対策	oad のホスト名の設定を確認してください。oad のホスト名のプロパティ名は org.omg.CORBA.ORBInitialHost です。

・マイナーコード 5085 (vmcid: 0x1000 minor code: 989)

原因	oad のポート番号が 0

対策	oad のポート番号の設定を確認してください。 oad のポート番号のプロパティ名は org.omg.CORBA.ORBInitialPort(jp.co.nec.orb.OadPort)です。

・マイナーコード 5086 (vmcid: 0x1000　minor code: 990)

原因	ローカルホスト名が不正
対策	ホスト名の設定を確認してください。

・マイナーコード 5173 (vmcid: 0x1000　minor code: 1077)

原因	SSL の初期化に失敗した
対策	SSL 関連の設定が正しいか、SSL に必要なライブラリが CLASSPATH に含まれているかを確認してください。

MARSHAL

・マイナーコード 5001 (vmcid: 0x1000　minor code: 905)

原因	メッセージ ヘッダ内に不正な Magic 識別子があった
対策	ルータやハブなどにおいて、ネットワーク障害が発生していないかを確認してください。

・マイナーコード 5002 (vmcid: 0x1000　minor code: 906)

原因	メッセージ ヘッダ内に未サポートの GIOP バージョンが指定されていた
対策	ルータやハブなどにおいて、ネットワーク障害が発生していないかを確認してください。

・マイナーコード 5003 (vmcid: 0x1000　minor code: 907)

原因	メッセージ ヘッダ内に未サポートの IIOP バージョンが指定されていた
対策	ルータやハブなどにおいて、ネットワーク障害が発生していないかを確認してください。

・マイナーコード 5006 (vmcid: 0x1000　minor code: 910)

原因	受信したメッセージの読み込みに失敗した
対策	サーバとクライアントのインターフェース(IDL)の定義が合っていない可能性があります。サーバもしくはクライアントのどちらか一方のファイルを更新して、アプリケーションのファイルのバージョンが合っていない状態となっていないかを確認してください。

・マイナーコード 5007 (vmcid: 0x1000　minor code: 911)

原因	java.io.IOException が発生した
対策	サーバマシンもしくは WebOTX のシステム/サービスが起動しているかを確認してください。

・マイナーコード 5141 (vmcid: 0x1000　minor code: 1045)

原因	ユーザコードの実行中にランタイム例外が発生した
対策	サーバ側のログなどで発生した例外を確認してください。

・マイナーコード 5185 (vmcid: 0x1000　minor code: 1089)

原因	通信プロトコル不正
対策	クライアントとサーバの以下のプロパティの設定を確認してください。 jp.co.nec.orb.IIOPMinorNumber(jp.co.nec.orb.IIOPMinorVersion) jp.co.nec.orb.IIOPCompatible

・マイナーコード 5186 (vmcid: 0x1000　minor code: 941)

原因	通信プロトコル不正
対策	クライアントとサーバの以下のプロパティの設定を確認してください。 jp.co.nec.orb.IIOPMinorNumber(jp.co.nec.orb.IIOPMinorVersion) jp.co.nec.orb.IIOPCompatible

・マイナーコード 5206 (vmcid: 0x1000　minor code: 1110)

原因	java.lang.ClassNotFoundException が発生した
対策	使用しているクラスが CLASSPATH に含まれているかを確認してください。

・マイナーコード 5207 (vmcid: 0x1000　minor code: 1111)

原因	public default constructor を持たない Serializable をアンマーシャルしようとしたが、ospijni の DLL をロードできなかったため、アンマーシャルできなかった
対策	Object Broker Java™ は JNI を使用しています。Object Broker Java™ は次の JNI ライブラリをロードします。 Windows C:¥Program Files¥NEC¥WebOTX¥ObjectBroker¥bin¥ospijni.dll HP-UX (PA-RISC) /opt/ObjectSpinner/lib/libospijni.sl その他の UNIX /opt/ObjectSpinner/lib/libospijni.so Object Broker Java™ の実装クラス群を -Xbootclasspath オプションに指定して java コマンドを実行した場合、以下のいずれかの設定が必要です。 sun.boot.library.path システムプロパティに、JNI ライブラリが格納されているディレクトリを追加する（sun.boot.library.path システムプロパティには、その JavaVM が必要とする JNI ライブラリのディレクトリをすべて指定する必要があります） JavaVM が必要とする JNI ライブラリを以下のディレクトリにコピーする（UNIX の場合はシンボリックリンクでも可） Windows <JDK>¥jre¥bin (JRE の場合は <JRE>¥<VERSION>¥bin) UNIX <JDK>/jre/lib/<HW> (JRE の場合は <JRE>/lib/<HW>) ※<JDK>は使用する JDK のインストールディレクトリ、<JRE>は使用する JRE のインストールディレクトリ、<VERSION>は使用する JRE のバージョンを表します。<HW>はハードウェアによってディレクトリ名が異なります。

・マイナーコード 5208 (vmcid: 0x1000　minor code: 1112)

原因	final フィールドを持つ Serializable をアンマーシャルしようとしたが、ospijni の DLL をロードできなかったため、アンマーシャルできなかった
対策	マイナーコード 5207 (vmcid: 0x1000　minor code: 1111)の 対策 を参照してください。

・ マイナーコード 1330446337 (vmcid: OMG　minor code: 1)

原因が複数考えられます。

原因	value ファクトリが見つからない
対策	使用しているクラスが CLASSPATH に含まれているかを確認してください。

原因	クライアントとサーバで JDK のバージョンが一致していない
対策	クライアントとサーバで JDK のバージョンが一致しているかを確認してください。

OBJ_ADAPTER

・マイナーコード 5106 (vmcid: 0x1000　minor code: 1010)

原因	POA マネージャが inactive 状態である
対策	POA マネージャが活性化されているか確認してください。

・マイナーコード 5179 (vmcid: 0x1000　minor code: 1080)

原因	指定された ServantManager が不正
対策	RETAIN ポリシであれば ServantActivator、NON_RETAIN ポリシであれば ServantLocator インタフェースを実装した ServantManager を利用してください。

・マイナーコード 1330446337 (vmcid: OMG　minor code: 1)

原因	POA がアクティブでない
対策	POA が活性化されているか確認してください。

OBJECT_NOT_EXIST

・マイナーコード 5112 (vmcid: 0x1000　minor code: 1016)

原因	オブジェクトが活性化されていない
対策	サーバオブジェクトが活性化されているか確認してください。

NO_IMPLEMENT

・マイナーコード 1330446337 (vmcid: OMG　minor code: 1)

原因	ローカルな value 実装が見つからない
対策	使用しているクラスが CLASSPATH に含まれているかを確認してください。

INV_OBJREF

・マイナーコード 5137 (vmcid: 0x1000　minor code: 1041)

原因	コードセットの設定が正しくない
対策	サポートされているコードセットを正しく設定してください。コードセットの設定については、「アプリケーション開発ガイド 第 4 部 7 章」-「7.2.7 Object Broker Java の機能」-「7.2.7.8 文字コードセット」を参照してください。

BAD_PARAM

・マイナーコード 5018 (vmcid: 0x1000　minor code: 922)

原因	要素の範囲外のインデックスが指定された
対策	定義した配列などの大きさと、引数などに指定した値を確認してください。

・マイナーコード 5034 (vmcid: 0x1000　minor code: 938)

原因	指定されたサーバオブジェクトはインターフェースをサポートしていない
対策	オブジェクトとインターフェースの型が一致しているかを確認してください。

・マイナーコード 5187 (vmcid: 0x1000　minor code: 1091)

原因	不正な valuetype（StreamableValue、CostomValue 以外）が指定された
対策	valuetype の実装を確認してください。

・マイナーコード 5241 (vmcid: 0x1000　minor code: 1145)

原因	指定した全ての名前サーバにリファレンスが登録されていない
対策	接続先の名前サーバおよび CNS の設定を確認してください。また、リファレンスが登録されているかどうかを確認してください。

・マイナーコード 5242 (vmcid: 0x1000　minor code: 1146)

原因	指定した名前サーバにリファレンスが登録されていない
対策	接続先の名前サーバおよび CNS の設定を確認してください。また、リファレンスが登録されているかどうかを確認してください。

・マイナーコード 5243 (vmcid: 0x1000　minor code: 1147)

原因	複数の名前サーバに対する要求で異なるシステム例外が発生した
対策	名前サーバおよび CNS が正常に起動しているかを確認してください。

・マイナーコード 1330446344 (vmcid: OMG　minor code: 8)

原因	指定された URL アドレスが正しくない
対策	URL に記述したホスト名、ポート番号、サービス名、オブジェクト名が正しいかどうか、名前サービスもしくは CNS が正常に起動しているかを確認してください。

NO_PERMISSION

・マイナーコード 5228 (vmcid: 0x1000　minor code: 1132)

原因が複数考えられます。

原因	CSIv2 のアクセス判定で否認された
対策	指定した認証情報の内容を確認してください。

原因	CSIv2 のコールバックが正常に登録されていない
対策	コールバックが正常に登録されているか確認してください。

BAD_OPERATION

・マイナーコード 5041 (vmcid: 0x1000　minor code: 945)

原因	このオペレーションは存在しない
対策	クライアントとサーバでインターフェース(IDL)の定義が合っているかを確認してください。

・マイナーコード 5256 (vmcid: 0x1000　minor code: 1160)

原因	リフレクションで構築した Tie のメソッドのデータの中に、呼び出すべきメソッドが見つからない

対策	EJB サーバ側のリモートインタフェースの定義がクライアント側のものと同じかを確認してください。

3.3.2. Object Broker C++例外一覧

Object BrokerC++で発生する例外ごとに、マイナーコード別の原因と対策を一覧にしています。

BAD_PARAM

・マイナーコード 1025

原因	ホスト名が正しくない。あるいは hosts ファイル、DNS などで IP アドレスを知ることができない
対策	オブジェクトリファレンスに含まれているホスト名がクライアント側で IP アドレスに変換できていない可能性があります。ネットワークの設定を確認してください。 オブジェクトリファレンスには"マシン名.ドメイン名"で登録されている状態で、クライアントを起動したホストが"マシン名"でなければ変換できない設定になっているなどが考えられます。

COMM_FAILURE

・マイナーコード 1186

原因	コネクションがリセットされた(ECONRESET)
対策	高負荷により connect に失敗している可能性があります。 サーバ側のオプション設定で ListenBackLog の値を大きくしてください。

INV_OBJREF

・マイナーコード 1036

原因	accept 用ソケット作成に失敗した。
対策	ポート番号が他のプロセスで使用されていないかを確認してください。問題ない場合、ホスト名が正しく指定されているか、DNS などによるホスト名解決で正しく検索可能かを確認してください。

NO_PERMISSION

・マイナーコード 1159

原因	要求を行ったクライアントホストからのサーバプロセスの自動起動は禁止されている
対策	クライアントマシンからの自動起動が許可されていません。 サーバマシンの oad の設定を変更してください(actallow ファイルにクライアントマシン名を追加します)。

・マイナーコード 1160

原因	このサーバプログラムの自動起動は禁止されている
対策	ObjectBroker は、conf ディレクトリの下にある exeallow というファイルなどによって自動起動を制限することができます。設定を変更してください。 自動起動についての設定をしていない場合にはサーバが異常終了している可能性があります。 該当するサーバプロセスの確認を行い、サーバが終了している場合には再起動を行ってください。

・マイナーコード 1161

原因	自動起動サーバで相対パスは許されていない SSL サーバへの接続時にも発生する場合あり (ポート番号が未指定のサーバの場合)サーバが起動しているかを確認してください)

対策	サーバへのパスに"/../"を含んでいる可能性があります。オプション設定の OadAllowRelativePath に true を設定すれば回避できますが、絶対パスによる指定に変更することをおすすめします。 パスの指定方法を間違っている可能性があります。たとえば Windows 上で動作するオブジェクトに"/dir1/dir2/servername"のようなパスを指定した可能性があります。パスの指定方法が合っているか確認してください。

NO_RESPONSE

・マイナーコード 1027

原因	コネクションが失われた。ネットワーク不通、サーバホストダウン、サーバプロセス終了など
対策	活性化通知後、サーバがダウンしたか、もしくは、終了した可能性があります。パーシステントサーバならば再起動してください。ソケットを再利用する設定になっていた場合、サーバ終了後にクライアントが行った最初のオブジェクト呼び出しがエラーになります。この場合、再度呼び出すと正常に通信が行われます。

・マイナーコード 1080

原因	指定されたホスト上にインプリメンテーションが存在しない。あるいは、ブロードキャストを用いて探したけれどもインプリメンテーションが見つからなかった。
対策	指定されたホスト上で名前サーバが起動しているか確認して下さい。

・マイナーコード 1141

原因	コネクションの作成でタイムアウトした
対策	呼び出したパーシステントサーバを起動していない可能性があります。サーバが動作しているか確認してください。 サーバが起動しているかを確認してください。起動している場合は、コネクションタイムアウトの設定を変更してください。

・マイナーコード 1148

原因	recv でタイムアウトした
対策	ネットワークの設定が間違っている可能性があります。ホスト名からIP アドレスが参照できるようにネットワーク(/etc/hosts 等)の設定を行ってください。 サーバの処理が滞っていないか、またはサーバが動作しているかを確認してください。問題がなければ、タイムアウト時間の設定を変更してください。

・マイナーコード 1186

原因	コネクションがリセットされた(ECONRESET)
対策	send や recv で WSAECONNRESET 例外が発生しました。活性化通知後、サーバがダウンしたか、もしくは、終了した可能性があります。パーシステントサーバならば再起動してください。ソケットを再利用する設定になっている場合、サーバ終了後にクライアントが行った最初のオブジェクト呼び出しがエラーになります。この場合、再度呼び出すと正常に通信が行われます。

TRANSIENT

・マイナーコード 1211

原因	サーバがコネクションをクローズした
対策	サーバからの CloseConnection メッセージを受信しました。 サーバが停止している可能性があります。

OBJ_ADAPTER

・マイナーコード 1140

原因	サーバの起動に失敗した
対策	自動起動サーバの実行ファイルがない可能性があります。実行ファイル削除したり、ディレクトリや実行ファイルの名前を変更していないか確認してください。 インプリメンテーション登録時に、自動起動サーバの実行ファイルへのパスを間違えた可能性があります。パスが間違っていないか確認してください。間違っている場合は登録しなおしてください。

3.4. JavaVMのスレッドダンプ取得

WebOTX が起動中にストール状態になる、または、コマンドからの応答がなく停止できない状態など、WebOTX が無応答状態に陥った場合には以下の手順で JavaVM のスレッドダンプを取得し開発元に問い合わせを行ってください。

3.4.1. UNIXのスレッドダンプ

以下の手順でスレッドダンプを採取します。

取得方法

1. ドメインの JavaVM のプロセス番号を特定します。
   ps コマンドによりドメインの JavaVM のプロセス番号を特定します。ps コマンドの結果では、他の JavaVM プロセスとの区別がつきにくいため JavaVM に指定された-X オプションを手がかりにプロセスの特定を行ってください。

   ps -exf |grep java |grep ドメイン名 |grep funcid=agent

   ドメインの JavaVM にはデフォルトインストールで以下のオプションが引数に指定されています。
   -server -Dwebotx.funcid=agent -Ddomain.name=domain1 -Xms64m -Xmx512m -XX:NewRatio=2
2. 1で特定したプロセス番号に対して kill -SIGQUIT を実行します。
   無限ループによる障害を検出するために、数秒間隔で複数回実行するようにしてください。

   kill -SIGQUIT <PID>

3. 以下のファイルにスレッドダンプの結果が出力されます。

   /opt/WebOTX/domains/ドメイン名/logs/server.log

(注)プロセス番号(<PID>)はプロセスを起動するたびに異なる値が割り振られます。そのため、ヒープダンプの採取時には必ず ps コマンドを実行し、指定するプロセス番号が現在実行中のドメインの JavaVM プロセスのプロセス番号であることを確認してください。

3.4.2. Windowsのスレッドダンプ

Windows では、JavaVM プロセスに対して Ctrl+Break キーを送信することでスレッドダンプが出力されます。
しかし、WebOTX はサービスとして起動されているために通常ではスレッドダンプの取得は行えません。スレッドダンプを出力させるようにするには以下の設定を行うことで取得できます。

ドメインを再起動する必要があるため再現性のある問題に対して有効です。ただし、この設定をすることによりドメインの起動時にデスクトップ上にコンソールが表示されます。 この状態でログオフを行うと Windows の仕様によりドメインプロセスが停止してしまいます。そのため、調査中はログオフを行わないようにしてください。また、障害調査以外の通常運用時は設定を元にもどすようにしてください。

(注)タスクスケジューラから WebOTX を起動する運用の場合、Windows の仕様によりコンソールが表示されないことがあります。 この場合は、Windows 標準の「AT」コマンド(「interactive」オプション指定が必要)での代用をご検討ください。

(注)この設定を行うと、エージェントプロセスで server.log に出力していたメッセージが以下のファイルに出力されるようになります。
　標準出力：${INSTANCE_ROOT}¥logs¥jvm_stdout.log
　標準エラー出力：${INSTANCE_ROOT}¥logs¥jvm_stderr.log
このファイルはドメインを再起動すると上書きされるため、出力されたメッセージを残す必要がある場合はドメインの再起動前にファイル名を変更するなどで退避してください。

設定方法

1. WebOTX を停止した状態で設定を行います。
   起動中にストールしコマンドを受け付けない状態になる場合は、一旦 "WebOTX AS Agent Service" のサービスを自動から手動に切り替えてサーバを再起動します。

   スタート→コントロールパネル→パフォーマンスとメンテナンス→管理ツール→サービス
   "WebOTX AS Agent Service" を自動から手動に変更

2. ドメインの設定に JavaVM オプションを追加します。

   <<J2SE 1.4、J2SE 5.0 の場合>>
   各ドメインの設定ファイル${INSTANCE_ROOT}¥config¥domain.xml に-Xrunwojconsole の記述を追加します。

   ```
   ～
   <jvm-options>-Xms64m</jvm-options>
   <jvm-options>-Xmx512m</jvm-options>
   <jvm-options>-Xrunwojconsole</jvm-options>　←この行を追加
   <jvm-options>-XX:NewRatio=2</jvm-options>
   ～
   ```

   <<J2SE 6.0 の場合>>
   各ドメインの設定ファイル${INSTANCE_ROOT}¥config¥domain.xml に-agentlib:wojconsole の記述を追加します。

   ```
   ～
   <jvm-options>-Xms64m</jvm-options>
   <jvm-options>-Xmx512m</jvm-options>
   <jvm-options>-agnetlib:wojconsole</jvm-options>　←この行を追加
   <jvm-options>-XX:NewRatio=2</jvm-options>
   ～
   ```

3. サービスのデスクトップとの会話を変更
   サービスのコントロールパネルを開き、WebOTX AS Agent Service の"デスクトップとの対話をサービスに許可"にチェックを入れます。

   スタート→コントロールパネル→パフォーマンスとメンテナンス→管理ツール→サービス
   WebOTX AS Agent Service→ログオン→デスクトップとの対話をサービスに許可

4. WebOTX AS Agent Service サービスを起動します。
   WebOTX が起動すると、デスクトップ画面上にコンソール画面が表示されるようになります。

5. 表示されたコンソール画面に、Ctrl+Break キーを入力します。
   無限ループによる障害を検出するために、数秒間隔で複数回実行するようにしてください。

6. 以下のファイルにスレッドダンプの結果が出力されます。

${AS_INSTALL}¥domains¥ドメイン名¥logs¥jvm_stdout.log

設定解除方法

1. WebOTX を停止した状態で設定を行います。
2. ドメインの設定から追加した JavaVM オプションを削除します。

   <<J2SE 1.4、J2SE 5.0 の場合>>
   各ドメインの設定ファイル${INSTANCE_ROOT}¥config¥domain.xml から-Xrunwojconsole の記述を削除します。

   ～
   <jvm-options>-Xms64m</jvm-options>
   <jvm-options>-Xmx512m</jvm-options>
   <jvm-options>-Xrunwojconsole</jvm-options>　←この行を削除
   <jvm-options>-XX:NewRatio=2</jvm-options>
   ～

   <<J2SE 6.0 の場合>>
   各ドメインの設定ファイル${INSTANCE_ROOT}¥config¥domain.xml から-agentlib:wojconsole の記述を削除します。

   ～
   <jvm-options>-Xms64m</jvm-options>
   <jvm-options>-Xmx512m</jvm-options>
   <jvm-options>-agentlibwojconsole</jvm-options>　←この行を削除
   <jvm-options>-XX:NewRatio=2</jvm-options>
   ～

3. サービスのデスクトップとの会話を変更
   サービスのコントロールパネルを開き、WebOTX AS Agent Service の"デスクトップとの対話をサービスに許可"からチェックをはずします。

   スタート→コントロールパネル→パフォーマンスとメンテナンス→管理ツール→サービス
   "WebOTX AS Agent Service"→ログオン→デスクトップとの対話をサービスに許可
4. WebOTX AS Agent Service サービスを起動します。

3.5. JavaVMのヒープダンプ取得

WebOTX の処理が遅く server.log に Full GC の行が多数出力されていた、または、WebOTX が異常終了し server.log に OutOfMemoryError が出力されていた場合には、以下の手順で JavaVM のヒープダンプを取得し開発元に問い合わせを行ってください。

(注)ヒープダンプは採取対象の JavaVM プロセスが現在使用しているヒープサイズ分の容量になります。最大ヒープサイズを大きな値に設定している場合は出力されるヒープダンプのサイズも大きくなりますので、取得の際には保存先の空き容量にご注意ください。

3.5.1. UNIXのヒープダンプ

以下の手順でヒープダンプを採取します。

取得方法

<<J2SDK 1.4、JDK 5.0 の場合>>

1. WebOTX を停止した状態で設定を行います。
2. ドメインの設定に JavaVM オプションを追加します。

   ドメインの設定ファイル${INSTANCE_ROOT}¥config¥domain.xml に-XX:+HeapDumpOnCtrlBreak の記述を追加します。

   ～
   <jvm-options>-Xms64m</jvm-options>
   <jvm-options>-Xmx512m</jvm-options>
   <jvm-options>-XX:+HeapDumpOnCtrlBreak</jvm-options>　←この行を追加
   <jvm-options>-XX:NewRatio=2</jvm-options>
   ～

   ドメインを起動して上記の変更を反映します。
3. ドメインの JavaVM のプロセス番号を特定します。
   ps コマンドによりドメインの JavaVM のプロセス番号を特定します。ps コマンドの結果では、他の JavaVM プロセスとの区別がつきにくいため JavaVM に指定された-X オプションを手がかりにプロセスの特定を行ってください。

   ps -exf |grep java |grep ドメイン名 |grep funcid=agent

   ドメインの JavaVM にはデフォルトインストールで以下のオプションが引数に指定されています。
   -server -Dwebotx.funcid=agent -Ddomain.name=<ドメイン名> -Xms64m -Xmx512m -XX:NewRatio=2

4. 3 で特定したプロセス番号に対して kill -SIGQUIT を実行します。

   kill -SIGQUIT <PID>

5. 以下のファイルにヒープダンプが出力されます。

   ${AS_INSTALL}¥domains¥ドメイン名¥config¥java_pid<PID>.hprof.<文字列>

(注)この方法を使う場合、WebOTX で使用している JDK(J2SDK)のバージョンが、j2sdk 1.4.2_12 以上または jdk5u15 以上である必要があります。

(注)プロセス番号(<PID>)はプロセスを起動するたびに異なる値が割り振られます。そのため、ヒープダンプの採取時には必ず ps コマンドを実行し、指定するプロセス番号が現在実行中のドメインの JavaVM プロセスのプロセス番号であることを確認してください。

<<JDK5.0、JDK 6.0 の場合>>

1. ドメインの JavaVM のプロセス番号を特定します。
   ps コマンドによりドメインの JavaVM のプロセス番号を特定します。ps コマンドの結果では、他の JavaVM プロセスとの区別がつきにくいため JavaVM に指定された-X オプションを手がかりにプロセスの特定を行ってください。

   ps -exf |grep java |grep ドメイン名 |grep funcid=agent

   ドメインの JavaVM にはデフォルトインストールで以下のオプションが引数に指定されています。
   -server -Dwebotx.funcid=agent -Ddomain.name=<ドメイン名> -Xms64m -Xmx512m -XX:NewRatio=2
2. JDK に付属の jmap で以下のコマンドを実行します。

   JDK5 の場合:jmap -heap:format=b <PID>
   JDK6 の場合:jmap -dump:format=b,file=<filename> <PID>

3. 2 のコマンドを実行したディレクトリに

   JDK5 の場合:heap.bin
   JDK6 の場合:<filename>に指定した文字列

   をファイル名としてヒープダンプのファイルが作成されます。

(注)プロセス番号(<PID>)はプロセスを起動するたびに異なる値が割り振られます。そのため、ヒープダンプの採取時には必ず ps コマンドを実行し、指定するプロセス番号が現在実行中のドメインの JavaVM プロセスのプロセス番号であることを確認してください。

設定解除方法

<<J2SDK 1.4、JDK 5.0 の場合>>

1. WebOTX を停止した状態で設定を行います。
2. ドメインの設定から追加した javaVM オプションを削除します。

   ドメインの設定ファイル${INSTANCE_ROOT}¥config¥domain.xml から-XX:+HeapDumpOnCtrlBreak の記述を削除します。

```
～
<jvm-options>-Xms64m</jvm-options>
<jvm-options>-Xmx512m</jvm-options>
<jvm-options>-XX:+HeapDumpOnCtrlBreak</jvm-options>   ←この行を削除
<jvm-options>-XX:NewRatio=2</jvm-options>
～
```

3. WebOTX を起動します。

<<JDK5.0、JDK 6.0 の場合>>

jmap を使用した手順では、設定解除作業は不要です。

3.5.2. Windowsのヒープダンプ

以下の手順でヒープダンプを採取します。

取得方法

(注) WebOTXのJavaVMプロセスに対してコマンドの実行を行うにはMicrosoft社提供のPsExecを使用します。
PsExecは以下のサイトから取得できます。事前にダウンロードしてマシンに配置してください。
PsExec(http://technet.microsoft.com/ja-jp/sysinternals/bb897553.aspx)
(注)以降の説明ではpsexec.exeを C:¥PSTools以下に配置したとして説明します。

<<J2SDK 1.4、JDK 5.0 の場合>>

(注)Windowsでは、JavaVMプロセスに対して-XX:+HeapDumpOnCtrlBreakオプションを指定しCtrl+Breakを送信することでヒープダンプが出力されます。Ctrl+Breakを送信する手段としてフリーソフトのSendSignalを使用した方法を説明します。フリーソフトが使用できない場合は、3.4.2. Windowsのスレッドダンプで説明しているコンソールからCtrl+Breakを送信する方法を使用してください。
SendSignalは以下のサイトから取得できます。事前にダウンロードしてマシンに配置してください。
SendSignal(http://www.latenighthacking.com/projects/2003/sendsignal/)
(注)以降の説明ではsendsignal.exeをCドライブ直下に配置したとして説明します。

1. WebOTX を停止した状態で設定を行います。

起動中にストールしコマンドを受け付けない状態になる場合は、一旦 "WebOTX AS Agent Service" のサービスを自動から手動に切り替えてサーバを再起動します。

スタート→コントロールパネル→パフォーマンスとメンテナンス→管理ツール→サービス
"WebOTX AS Agent Service" を自動から手動に変更

2. ドメインの設定に javaVM オプションを追加します。

ドメインの設定ファイル${INSTANCE_ROOT}¥config¥domain.xml に-XX:+HeapDumpOnCtrlBreak の記述を追加します。

```
〜
<jvm-options>-Xms64m</jvm-options>
<jvm-options>-Xmx512m</jvm-options>
<jvm-options>-XX:+HeapDumpOnCtrlBreak</jvm-options>   ←この行を追加
<jvm-options>-XX:NewRatio=2</jvm-options>
〜
```

ドメインを起動して変更を反映します。

3. ドメインの JavaVM のプロセス番号を特定します。
jpsコマンドによりドメインの JavaVMのプロセス番号を特定します。コマンドプロンプトから以下のコマンドを実行してください。

C:¥PSTools>psexec.exe -s "<JAVA_HOME>¥bin¥jps" -v

この時出力された結果のうち、以下の行の PID が今回対象とするプロセスのプロセス番号です。
<PID> ChildMain

jps コマンドの結果では、他の JavaVM プロセスとの区別がつきにくいため JavaVM に指定された-X オプションを手がかりにプロセスの特定を行ってください。
ドメインの JavaVM にはデフォルトインストールで以下のオプションが引数に指定されています。
-Dwebotx.funcid=agent -Ddomain.name=<ドメイン名> -Xms64m -Xmx512m -XX:NewRatio=2

4. 3 で特定したプロセス番号に対して sendsignal を実行します。

C:¥PSTools>psexec.exe -s "C:¥sendsignal.exe" <PID>

5. 以下のファイルにヒープダンプの結果が出力されます。

${AS_INSTALL}¥domains¥ドメイン名¥config¥java_pid<PID>.hprof.<文字列>

(注)この方法を使う場合、WebOTX で使用している JDK(J2SDK)のバージョンが、j2sdk 1.4.2_12 以上または jdk5u15 以上である必要があります。

(注)プロセス番号(<PID>)はプロセスを起動するたびに異なる値が割り振られます。そのため、ヒープダンプの採取時には必ず jps コマンドを実行し、指定するプロセス番号が現在実行中のドメインの JavaVM プロセスのプロセス番号であることを確認してください。

<<JDK 6.0 の場合>>

1. ドメインの JavaVM のプロセス番号を特定します。
jps コマンドによりドメインの JavaVM のプロセス番号を特定します。

C:¥PSTools>psexec.exe -s "<JAVA_HOME>¥bin¥jps" -v

この時出力された結果のうち、以下の行の PID が今回対象とするプロセスのプロセス番号です。
<PID> ChildMain

jps コマンドの結果では、他の JavaVM プロセスとの区別がつきにくいため JavaVM に指定された-X オプションを手がかりにプロセスの特定を行ってください。

ドメインの JavaVM にはデフォルトインストールで以下のオプションが引数に指定されています。
-Dwebotx.funcid=agent -Ddomain.name=<ドメイン名> -Xms64m -Xmx512m -XX:NewRatio=2

2. JDK に付属の jmap で以下のコマンドを実行します。

   C:¥PSTools>psexec.exe -s "<JAVA_HOME>¥bin¥jmap" -dump:format=b,file=<filename> <PID>

3. 以下のフォルダに<filename>で指定した名前でファイルが作成されます。

   C:¥WINDOWS¥system32

(注)プロセス番号(<PID>)はプロセスを起動するたびに異なる値が割り振られます。そのため、ヒープダンプの採取時には必ず jps コマンドを実行し、指定するプロセス番号が現在実行中のドメインの JavaVM プロセスのプロセス番号であることを確認してください。

設定解除方法

<<J2SDK 1.4、JDK 5.0 の場合>>

1. WebOTX を停止した状態で設定を行います。
2. ドメインの設定から追加した JavaVM オプションを削除します。

   ドメインの設定ファイル${INSTANCE_ROOT}¥config¥domain.xml から-XX:+HeapDumpOnCtrlBreak の記述を削除します。

   ```
   ～
   <jvm-options>-Xms64m</jvm-options>
   <jvm-options>-Xmx512m</jvm-options>
   <jvm-options>-XX:+HeapDumpOnCtrlBreak</jvm-options>   ←この行を削除
   <jvm-options>-XX:NewRatio=2</jvm-options>
   ～
   ```

3. WebOTX を起動します。

<<JDK 6.0 の場合>>

jmap を使用した手順では、設定解除作業は不要です。

3.6. ドメインの強制停止

WebOTX エージェントプロセスが何らかの原因で異常終了した場合に、WebOTX で起動された各サービス(JMS、Web サーバ、TP モニタ・マネージャ、Transaction サービス、ObjectBroker サービス)のプロセスが残存することがあります。運用管理コマンドを使用し、以下の手順でドメインの強制停止を行ってください。

3.6.1. 停止方法

以下の手順でドメインを強制停止します。

強制停止方法

stop-domain の--force オプションを指定して、該当ドメインを停止してください(domain1 の部分がドメイン名を表します)。

例)

otxadmin stop-domain --force domain1

なお、--force オプション指定時のログは
${AS_INSTALL}/domains/domain1/logs/agent/webotx_prcstop.log に出力されます。

ただし、ドメインが runnning 状態で上記コマンドを実行した場合には、通常のドメイン停止が行われ、ログへの出力はしません。